

George Grätzer

Text and Math Into L^AT_EX

6th Edition

MOREMEDIA



Springer

Text and Math Into L^AT_EX

George Grätzer

Text and Math Into L^AT_EX

Sixth Edition

Foreword by Rainer Schöpf | LaTeX3 Team

 Springer

George Grätzer
Toronto, ON, Canada

ISBN 978-3-031-55280-9 ISBN 978-3-031-55281-6 (eBook)
<https://doi.org/10.1007/978-3-031-55281-6>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2007, 2016, 2024

Originally published with the title: More Math into LaTeX

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

To the young ones,

Emma (19),

Kate (17),

Jay (12)

Short Contents

Foreword	xxi
Preface to the sixth edition	xxv
Introduction	xxvii
I Mission Impossible	1
1 Short course	3
2 And one more thing	33
II Text into L^AT_EX	45
3 Typing text	47
4 Text environments	97
III Fonts for text and math	131
5 Font basics	133
6 Font encoding	141
IV Math into L^AT_EX	147
7 Typing math	149

8	More math	187
9	Multiline math displays	211
V	Document Structure	249
10	Documents	251
11	The AMS article document class	279
VI	PDF Documents	313
12	Hyperlinks	315
13	Presentations	321
14	Illustrations	357
VII	Customization	401
15	Commands and environments	403
VIII	Long Documents	463
16	BIB _{TEX}	465
17	Beyond BIB _{TEX}	491
18	<i>MakeIndex</i>	499
19	Books in L ^A _{TEX}	515
A	Math symbol tables	531
B	Text symbol tables	545
C	ChatGPT 101	551
	Bibliography	559
	Index	562

Contents

Foreword	xxi
Preface to the sixth edition	xxv
Introduction	xxvii
What this book about?	xxvii
Is this book for you?	xxviii
What's in the book?	xxix
Conventions	xxxi
I Mission Impossible	1
1 Short course	3
1.1 Getting started	5
1.1.1 Your \LaTeX	5
1.1.2 Sample files	5
1.1.3 Editing cycle	5
1.1.4 Typing the source file	6
1.2 The keyboard	7
1.3 Your first text note	8
1.4 Lines too wide	11
1.5 A note with formulas	12
1.6 The building blocks of a formula	14
1.7 Displayed formulas	19
1.7.1 Equations	19
1.7.2 Symbolic referencing	20
1.7.3 Aligned formulas	21
1.7.4 Cases	24

1.8	The anatomy of a document	25
1.9	Your own commands	27
1.10	Adding an illustration	27
1.11	The anatomy of a presentation	28
2	And one more thing	33
2.1	Structure	33
2.2	Auxiliary files	34
2.3	Logical and visual design	37
2.4	General error messages	37
2.5	Errors in math	40
2.6	Your errors: Davey's Dos and Don'ts	41
II	Text into L^AT_EX	45
3	Typing text	47
3.1	The keyboard	48
3.1.1	Basic keys	48
3.1.2	Special keys	49
3.1.3	Prohibited keys	49
3.2	Words, sentences, and paragraphs	50
3.2.1	Spacing rules	50
3.2.2	Periods	52
3.3	Commanding L ^A T _E X	54
3.3.1	Commands and environments	54
3.3.2	Scope	58
3.3.3	Types of commands	60
3.4	Symbols not on the keyboard	61
3.4.1	Quotation marks	61
3.4.2	Dashes	61
3.4.3	Ties or nonbreakable spaces	62
3.4.4	Special characters	63
3.4.5	Ellipses	64
3.4.6	Ligatures	64
3.4.7	Accents and symbols in text	65
3.4.8	Logos and dates	66
3.4.9	Hyphenation	68
3.5	Comments and footnotes	71
3.5.1	Comments	71
3.5.2	Footnotes	74
3.6	Lines, paragraphs, and pages	75
3.6.1	Lines	75

3.6.2	Paragraphs	79
3.6.3	Pages	80
3.6.4	Multicolumn printing	81
3.7	Spaces	81
3.7.1	Horizontal spaces	81
3.7.2	Vertical spaces	83
3.7.3	Relative spaces	85
3.7.4	Expanding spaces	85
3.8	Boxes	86
3.8.1	Line boxes	86
3.8.2	Frame boxes	90
3.8.3	Paragraph boxes	91
3.8.4	Marginal comments	93
3.8.5	Solid boxes	94
3.8.6	Fine tuning boxes	95
4	Text environments	97
4.1	Some general rules	98
4.2	List environments	98
4.2.1	Numbered lists	98
4.2.2	Bulleted lists	99
4.2.3	Captioned lists	100
4.2.4	A rule and combinations	100
4.3	Style and size environments	104
4.4	Proclamations (theorem-like structures)	105
4.4.1	The full syntax	108
4.4.2	Proclamations with style	109
4.5	Proof environments	111
4.6	Tabular environments	113
4.6.1	Table styles	120
4.7	Tabbing environments	121
4.8	Miscellaneous displayed text environments	123
III	Fonts for text and math	131
5	Font basics	133
5.1	Shape, size, serif	133
5.2	Document font families	134
5.3	Shape commands	136
5.3.1	Italic corrections	137
5.3.2	Series	138
5.4	Size changes	138

5.5	Orthogonality	139
5.6	Obsolete two-letter commands	140
6	Font encoding	141
6.1	OT1, T1,	141
6.2	Low-level commands	142
6.3	PostScript fonts	143
6.4	L ^A T _E X localized	145
IV	Math into L^AT_EX	147
7	Typing math	149
7.1	Math environments	150
7.2	Spacing rules	152
7.3	Equations	153
7.4	Basic constructs	154
7.4.1	Arithmetic operations	155
7.4.2	Binomial coefficients	157
7.4.3	Ellipses	157
7.4.4	Integrals	159
7.4.5	Roots	159
7.4.6	Text in math	160
7.4.7	Hebrew and Greek letters	162
7.5	Delimiters	163
7.5.1	Stretching delimiters	164
7.5.2	Delimiters that do not stretch	165
7.5.3	Limitations of stretching	165
7.5.4	Delimiters as binary relations	167
7.6	Operators	167
7.6.1	Operator tables	167
7.6.2	Congruences	169
7.6.3	Large operators	169
7.6.4	Multiline subscripts and superscripts	170
7.7	Math accents	173
7.8	Stretchable horizontal lines	173
7.8.1	Horizontal braces	173
7.8.2	Overlines and underlines	175
7.8.3	Stretchable arrow math symbols	175
7.9	Building a formula step-by-step	176
7.10	Formula Gallery	178

8	More math	187
8.1	Spacing of symbols	187
8.1.1	Classification	188
8.1.2	Three exceptions	188
8.1.3	Spacing commands	190
8.1.4	Examples	190
8.1.5	The <code>phantom</code> command	191
8.2	The STIX math symbols	192
8.3	Building new symbols	194
8.3.1	Stacking symbols	194
8.3.2	Negating and side-setting symbols	197
8.3.3	Changing the type of a symbol	198
8.4	Math alphabets and symbols	199
8.4.1	Math alphabets	199
8.4.2	Math symbol alphabets	200
8.4.3	Bold math symbols	201
8.4.4	Size changes	203
8.4.5	Continued fractions	203
8.5	Vertical spacing	203
8.6	Tagging and grouping	205
8.7	Miscellaneous	207
9	Multiline math displays	211
9.1	Visual Guide	211
9.1.1	Columns	211
9.1.2	Subsidiary math environments	213
9.1.3	Adjusted columns	214
9.1.4	Aligned columns	214
9.1.5	Touring the Visual Guide	214
9.2	Gathering formulas	215
9.3	Splitting long formulas	216
9.4	Some general rules	219
9.4.1	General rules	219
9.4.2	Subformula rules	219
9.4.3	Breaking and aligning formulas	221
9.4.4	Numbering groups of formulas	222
9.5	Aligned columns	223
9.5.1	An <code>align</code> variant	225
9.5.2	<code>eqnarray</code> , the ancestor of <code>align</code>	226
9.5.3	The subformula rule revisited	227
9.5.4	The <code>alignat</code> environment	228
9.5.5	Inserting text	230

9.6	Aligned subsidiary math environments	231
9.6.1	Subsidiary variants	232
9.6.2	Split	234
9.7	Adjusted columns	236
9.7.1	Matrices	237
9.7.2	Arrays	241
9.7.3	Cases	243
9.8	Commutative diagrams	244
9.9	Adjusting the display	247

V Document Structure 249

10 Documents 251

10.1	The structure of a document	252
10.2	The preamble	253
10.3	Top matter	255
10.4	Main matter	255
10.4.1	Sectioning	255
10.4.2	Cross-referencing	259
10.4.3	Floating tables and illustrations	264
10.5	Back matter	268
10.5.1	Bibliographies in articles	268
10.5.2	Simple indexes	274
10.6	Visual design	275

11 The AMS article document class 279

11.1	Why <code>amsart</code> ?	279
11.1.1	Submitting an article to the AMS	279
11.1.2	Submitting an article to Algebra Universalis	280
11.1.3	Submitting to other journals	280
11.1.4	Submitting to conference proceedings	281
11.2	The top matter	281
11.2.1	Article information	281
11.2.2	Author information	283
11.2.3	AMS information	287
11.2.4	Multiple authors	288
11.2.5	Examples	289
11.2.6	Abstract	292
11.3	The sample article	293
11.4	Article templates	303
11.5	Options	306
11.6	The AMS packages	309

11.7 Legacy documents	312
---------------------------------	-----

VI PDF Documents 313

12 Hyperlinks 315

12.1 Hyperlinks for L ^A T _E X	315
12.1.1 Hyperlinks	315
12.1.2 Using hyperref	316
12.1.3 backref and colorlinks	316
12.1.4 Bookmarks	317
12.1.5 Additional commands	318
12.2 Line numbers	319

13 Presentations 321

13.1 Quick and dirty beamer	322
13.2 Baby beamers	327
13.2.1 Overlays	327
13.2.2 Understanding overlays	332
13.2.3 More on the \only and \onslide commands	334
13.2.4 Lists as overlays	335
13.2.5 Out-of-sequence slides	336
13.2.6 Blocks and slides	338
13.2.7 Links	338
13.2.8 Columns	342
13.2.9 Coloring	343
13.3 The structure of a presentation	346
13.3.1 Longer presentations	349
13.3.2 Navigation symbols	349
13.4 Notes	350
13.5 Themes	351
13.6 Planning your presentation	353
13.7 What did I leave out?	354

14 Illustrations 357

14.1 First steps: objects on the page	358
14.1.1 Straight lines	358
14.1.2 Parameters for lines	360
14.1.3 Circles, ellipses, dots, and parts thereof	362
14.1.4 The next step: curved lines between points	365
14.1.5 A few more predefined objects	369
14.2 Some tools of the trade	371
14.2.1 Transformations	371

14.2.2	A little repetition	374
14.2.3	Adding some color	380
14.3	Next Step: text and labels on the page	382
14.3.1	Labels, labels everywhere	382
14.3.2	Larger blocks of text	386
14.3.3	Let <code>TikZ</code> do the computations	387
14.4	Graphs of functions and presentations of data	393
14.5	A little planning goes a long way	394
14.5.1	An interesting example	395
14.6	Conclusions	398
14.6.1	<code>TikZ</code> manual	398
14.6.2	Vector graphics	398

VII Customization 401

15	Commands and environments	403
15.1	Custom commands	404
15.1.1	Examples and rules	404
15.1.2	Arguments	410
15.1.3	Short arguments	413
15.1.4	Optional arguments	414
15.1.5	Redefining commands	414
15.1.6	Defining operators	415
15.1.7	Redefining names	416
15.1.8	Showing the definitions of commands	418
15.1.9	Delimited commands	419
15.2	Custom environments	421
15.2.1	Modifying existing environments	422
15.2.2	Arguments	425
15.2.3	Optional arguments with default values	425
15.2.4	Short contents	426
15.2.5	Brand-new environments	426
15.3	A custom command file	427
15.4	The sample article with custom commands	436
15.5	Numbering and measuring	443
15.5.1	Counters	443
15.5.2	Length commands	447
15.6	Custom lists	451
15.6.1	Length commands for the <code>list</code> environment	451
15.6.2	The <code>list</code> environment	453
15.6.3	Two complete examples	455
15.6.4	The <code>trivlist</code> environment	458

15.7 The dangers of customization	459
---	-----

VIII Long Documents 463

16 BiBTeX 465

16.1 The database	467
16.1.1 Entry types	467
16.1.2 Typing fields	470
16.1.3 Articles	472
16.1.4 Books	473
16.1.5 Conference proceedings and collections	474
16.1.6 Theses	477
16.1.7 Technical reports	478
16.1.8 Manuscripts and other entry types	479
16.1.9 Abbreviations	480
16.2 Using BiBTeX	481
16.2.1 Sample files	481
16.2.2 Setup	482
16.2.3 Four steps of BiBT _E Xing	483
16.2.4 BiBT _E X files	485
16.2.5 BiBT _E X rules and messages	487
16.2.6 Submitting an article	489
16.2.7 Some resources	489

17 Beyond BiBTeX 491

17.1 BibL _A T _E X	491
17.1.1 Introduction	491
17.1.2 BibL _A T _E X overview	491
17.1.3 biber	492
17.1.4 Citations	492
17.1.5 My citations	495
17.1.6 Cross-referencing between entries	496
17.1.7 Sorting	496
17.2 Research tools	497

18 MakeIndex 499

18.1 Preparing the document	499
18.2 Index commands	503
18.3 Processing the index entries	509
18.4 Rules	511
18.5 Multiple indexes	513
18.6 Glossary	514

18.7	Index styles and packages	514
19	Books in L^AT_EX	515
19.1	Book document classes	516
19.1.1	Sectioning	516
19.1.2	Division of the body	517
19.1.3	Document class options	518
19.1.4	Title pages	518
19.2	Tables of contents, lists of tables and figures	519
19.2.1	Tables of contents	519
19.2.2	Lists of tables and figures	521
19.2.3	Exercises	522
19.3	Logical design	522
19.4	Final preparations for the publisher	525
19.5	If you create the PDF file for your book	527
A	Math symbol tables	531
A.1	Hebrew and Greek letters	531
A.2	Binary relations	533
A.3	Binary operations	536
A.4	Arrows	537
A.5	Delimiters	539
A.6	Operators	540
A.6.1	Large operators	542
A.7	Math accents and fonts	543
A.8	Math spacing commands	544
B	Text symbol tables	545
B.1	Some European characters	545
B.2	Text accents	546
B.3	Text font commands	546
B.3.1	Text font family commands	546
B.3.2	Text font size changes	547
B.4	Additional text symbols	548
B.5	Additional text symbols with T1 encoding	549
B.6	Text spacing commands	550
C	ChatGPT 101	551
C.1	Introduction	551
C.2	What is ChatGPT?	551
C.3	What it can't do	552
C.4	Asking good questions	553

C.5	The GPT Store	554
C.6	Style and grammar	554
C.7	TeX programming	555
C.7.1	Testing	557
C.8	Tokens, darn it	557
C.9	Conclusions	557
Bibliography		559
Index		562

Foreword

It was the autumn of 1989—a few weeks before the Berlin wall came down. George H. W. Bush was the American president and the American Mathematical Society (AMS) decided to outsource T_EX programming to Frank Mittelbach and me.

Why did the AMS outsource T_EX programming to us? This was, after all, a decade before the words “outsourcing” and “off-shore” entered the lexicon. There were many American T_EX experts. Why turn elsewhere?

For a number of years, the AMS tried to port the mathematical typesetting features of AMS-T_EX to L^AT_EX, but they made little progress with the AMSFonts. Frank and I had just published the New Font Selection Scheme (NFSS) for L^AT_EX, which went a long way to satisfy their needs. So it was logical that the AMS turned to us to add AMSFonts to L^AT_EX. Being young and enthusiastic, we convinced the AMS that the AMS-T_EX commands should be changed to conform to the L^AT_EX standards. Michael Downes was assigned as our AMS contact; his insight was a tremendous help.

We already had L^AT_EX-NFSS, which could be run in two modes: compatible with the old L^AT_EX or enabled with the new font features. We added the reworked AMS-T_EX code to L^AT_EX-NFSS, thus giving birth to AMS-L^AT_EX, released by the AMS at the 1990 meeting of the International Mathematical Union in Kyoto.

AMS-L^AT_EX was another variant of L^AT_EX. Many installations had several L^AT_EX variants to satisfy the needs of their users: with old and new font changing commands, with and without AMS-L^AT_EX, a single and a multi-language version. We decided to develop a Standard L^AT_EX that would reconcile all the variants. Out of a group of interested people grew what was later called the L^AT_EX3 team—and the L^AT_EX3 project got underway. The team’s first major accomplishment was the release of L^AT_EX2e in June 1994. This standard L^AT_EX incorporates all the improvements we wanted back in 1989. It is now very stable and it is uniformly used.

Under the direction of Michael Downes, our AMS-L^AT_EX code was turned into AMS packages that run under L^AT_EX just like other packages. Of course, the L^AT_EX3

team recognizes that these are special; we call them “required packages” because they are part and parcel of a mathematician’s standard toolbox.

Since then a lot has been achieved to make an author’s task easier. A tremendous number of additional packages are available today. The *L^AT_EX Companion*, 3rd edition¹, describes many of my favorite packages.

George Grätzer got involved with these developments in 1990, when he got his copy of AMS-*L^AT_EX* in Kyoto. The documentation he received explained that AMS-*L^AT_EX* is a *L^AT_EX* variant—read Lamport’s *L^AT_EX* book to get the proper background. AMS-*L^AT_EX* is not AMS-*T_EX* either—read Spivak’s AMS-*T_EX* book to get the proper background. The rest of the document explained in what way AMS-*L^AT_EX* differs from *L^AT_EX* and AMS-*T_EX*. Talk about a steep learning curve!

Luckily, George’s frustration working through this nightmare was eased by his lengthy e-mail correspondence with Frank and lots of telephone calls to Michael. Three years of labor turned into his first book on *L^AT_EX*, providing a “simple introduction to AMS-*L^AT_EX*”. This edition is more mature, but preserves what made his first book such a success. Just as in the first book, Part I, *Mission Impossible*, is a short introduction for the beginner. Chapter 1, *Short Course*, dramatically reduces the steep learning curve of a few weeks to a few hours in only 30 pages. Chapter 2, *And a few more things . . .* adds a few more advanced topics useful already at this early stage.

The rest of the book is a detailed presentation of everything you may need to know. George “teaches by example”. You find in this book many illustrations of even the simplest concepts. For articles, he presents the *L^AT_EX* source file and the typeset result. For formulas, he discusses the building blocks with examples, presents a *Formula Gallery*, and a *Visual Guide* for multiline formulas.

Going forth and creating “masterpieces of the typesetting art”—as Donald Knuth put it at the end of the *T_EX*book—requires a fair bit of initiation. This is the book for the *L^AT_EX* beginner as well as for the advanced user. You just start at a different point.

The topics covered include everything you need for mathematical publishing:

- Instructions on creating articles, from the simple to the complex.
- Converting an article to a presentation.
- Customize *L^AT_EX* to your own needs.
- The secrets of writing a book.
- Where to turn to get more information.

The many examples are complemented by a number of easily recognizable features:

Rules which you must follow.

Tips on what to be careful about and how to achieve some specific results.

¹This book references it as LC3.

Experiments to show what happens when you make mistakes—sometimes, it can be difficult to understand what went wrong when all you see is an obscure \LaTeX message.

This book teaches you how to convert your mathematical masterpieces into typographical ones, giving you a lot of useful advice on the way. How to avoid the traps for the unwary and how to make your editor happy. And you'll experience the fascination of doing it right. Using good typography to better express your ideas.

If you want to learn \LaTeX , buy this book and start with the *Short Course*. If you can have only one book on \LaTeX next to your computer, this is the one to have. And if you want to learn about the world of \LaTeX packages, also buy a second book, the *\LaTeX Companion* [current version: 3rd edition].

A handwritten signature in black ink that reads "Rainer Schöpf". The script is fluid and cursive, with the first letters of the first and last names being capitalized and prominent.

Rainer Schöpf
 \LaTeX 3 team

Preface to the Sixth Edition

I have changed the title of this edition to “Text and Math into L^AT_EX” to emphasize the importance of text. Accordingly, the old “Part II Text and Math” is now split into two parts.

L^AT_EX development is continuing, chronicled on the Web site

<https://www.latex-project.org/news/>

The new features mostly facilitate writing L^AT_EX packages.

Computers have become so much faster. When I started T_EXing, it took more than two minutes to typeset a page. Now this whole book typesets in less than 3 seconds. So gone is the section how to use the L^AT_EX commands `\include` and `\includeonly` to decrease the waiting time for typesetting a long document (LOL).

There were sections introducing PDF files and the Internet. These days, they are ubiquitous and well-known. These sections are also gone.

In earlier editions, T_EX files, B_IB_T_EX data bases, and index files were all ASCII files (the simplest form of text files containing only 95 printable characters). Since 2018, L^AT_EX has switched its default input encoding to UTF-8 (an extension of ASCII), so it accepts Grätzer. This change also affects B_IB_T_EX and MakeIndex. We’ll discuss this big change in Chapter 6.

Many sections are now enhanced by the discussion of useful new packages. The new Chapter 17 describes recent developments that enhance, or replace, B_IB_T_EX.

Two major new developments facilitated my work in writing this edition. First, the publication of *The L^AT_EX Companion. Third edition* [35] by F. Mittelbach with U. Fischer (we’ll refer to it as LC3). This book gives a pretty detailed overview of the L^AT_EX packages as of 2023. Second, ChatGPT, which has an incredible knowledge of all matters L^AT_EX. See my review [23] of LC3 and ChatGPT in Tugboat. (There is an almost unanimous opinion among L^AT_EX experts—including the author of LC3—that

ChatGPT “hallucinates and invents nonexistent commands and packages”. I have consulted ChatGPT hundreds of times and have checked the answers carefully. ChatGPT has not made a single mistake.)

To provide the readers with the same help from ChatGPT, I added Appendix C, ChatGPT 101 for L^AT_EX users.

My reliance on sample files to teach basic concepts has not changed. There are now more than 70 sample files, including the file `6colorpages` that reproduces the six color pages of this book in color.

I am very grateful to Barbara Beeton, Gregory L. Cherlin, David Derbes, and Murray Eisenberg for long lists of thoughtful corrections and suggestions. Here is one (bb): correct *economics, and linguistics*. The fact that there was minimal overlap among the hundreds of corrections provided by these four sources certainly gives one pause for reflection. . .

And thanks for Michael Doob for the new and improved TikZ chapter, Chapter 14.

Toronto, February, 2024

Introduction

What's this book about?

L^AT_EX!

If you want this formula in your L^AT_EX document

$$\sqrt{a^2 + b^2}$$

then type

`\sqrt{a^2 + b^2}`

The command `\sqrt` produces the square root (of width and height of the formula under it) and `^` is the exponentiation.

But, of course, a document with formulas is mostly text. So L^AT_EX has extensive facilities dedicated to producing text. Linguists use it for diacritical marks and high volume print shops to save on paper (TV Guide of old). L^AT_EX hyphenates in dozens of languages and handles right-to-left languages such as Arabic and Hebrew.

If you are interested in using L^AT_EX for text only, skip Sections 1.5–1.7 in Part I (the short course) and all of Part IV.

Before L^AT_EX, there was T_EX, initiated by D. Knuth in 1977. It was first released in 1978 (final version in 1989). L. Lamport created L^AT_EX as a collection of ‘macros’ to simplify the use of T_EX. This was first released in 1985.

The first version of L^AT_EX, known as LaTeX 2.09, was released in 1985. See the detailed history of T_EX and L^AT_EX in LC3. (See also my take on the subject in Appendix C of the fifth edition of this book.)

LaTeX is widely used in many fields, including mathematics, computer science, engineering, physics, chemistry, economics, and linguistics. (ChatGPT lists 45 fields.)

A quick overview of this book.

Part I provides a quick introduction to L^AT_EX, from typing examples of text and math to typing your first article such as on that shown on page 4 and similarly: creating your first presentation, such as the sample presentation (illustrated in Figure 1.5), in a very short time. The rest of the book provides a detailed exposition of L^AT_EX.

L^AT_EX has a huge collection of rules and commands. While the basics in Part I should serve you well in all your writing, most articles and presentations will also require to look up some special topics. Learn Part I well and become passingly familiar with the rest of the book, so when the need arises you know where to turn with your questions.

You can find specific topics in the Short Contents, the detailed Contents, and the Index.

Mathematicians find L^AT_EX very strange. A typical article in mathematics deals with a field defined by a few axioms, and the topic of the article needs only a few more. In contrast, L^AT_EX has hundreds of axioms—commands. We try to introduce you to these very gently, by starting with as few as possible. For instance, we introduce presentations with only five new commands.

What is document markup?

When you work with a word processor, you see your document on the computer monitor more or less as it looks when printed, with its various fonts, font sizes, font shapes (for instance, roman, italic) and weights (for instance, normal, boldface), interline spacing, indentation, and so on.

Working with a *markup language* is different. You type the *source file* of your article in a *text editor*, in which all characters appear in the same font. To indicate changes in the typeset text, you must add *text markup commands* to the source file. For instance, to emphasize the phrase `detailed description` in a L^AT_EX source file, type

```
\emph{detailed description}
```

The `\emph` command is a markup command. The marked-up text yields the typeset output

```
┌
└ detailed description
```

In order to typeset math, you need *math markup commands*. As a simple example, consider the formula $\int \sqrt{\alpha^2 + x^2} dx$. To mark it up in L^AT_EX, type

```
$\int \sqrt{\alpha^2 + x^2} dx$
```

You do not have to worry about determining the size of the integral symbol or how to construct the square root symbol that covers $\alpha^2 + x^2$. L^AT_EX does it all for you.

The three layers

The markup language we shall discuss comes in three layers: $\text{T}_{\text{E}}\text{X}$, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, and the AMS packages. Most $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ installations automatically place all three on your computer. You do not have to know what comes from which layer, so we consider the three together and call it $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.

The three platforms

Most of you run $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ on one of the following three computer types:

- A Windows computer, a computer running Microsoft Windows.
- A Mac, a Macintosh computer running macOS.
- A computer running UNIX, or a UNIX variant such as Solaris or Linux.

The $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ source file and the typeset version both look the same independent of what type of computer you have. However, the way you type your source file, the way you typeset it, and the way you view the typeset version depends on the computer and on the $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ implementation you use.

An outline of the book

Part I is *Mission Impossible*; it helps you to get started quickly with $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, to type your first articles and to make your first presentations; it also prepares you to tackle $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ in more depth in the subsequent parts.

Chapter 1 is the *Short Course*. You start writing your *first document*—as typeset on page 4—and prepare your *first presentation*—see some of the slides typeset on page 28. This chapter introduces how $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ uses the *keyboard* and how to *type text*. You do not need to learn much to understand the basics. Text markup is quite easy. You also learn math markup, which is not so straightforward.

Several sections in this chapter ease you into *mathematical typesetting*. There is a section on the basic building blocks of math formulas. Another one discusses equations. Finally, we present the two simplest multiline formulas, which should cover most of your everyday needs. Skip these if you are interested only in text into $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.

We also cover the elements of presentations with a simple example.

In **Chapter 2**, we explain how things work, the structure of $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, the auxiliary files, the logical and visual design of a document, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ error messages. And we present a long list of dos and don'ts, to help you write good $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.

Part II introduces *typing text* in $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.

Chapters 3 starts by covering the keyboard, special characters and accents, hyphenation, fonts, spacing, and boxes.

In **Chapter 4**, we discuss displayed text, including *lists* and *tables*, and *proclamations* (theorem-like structures)² and *proofs*.

Fonts for text and math are discussed in **Part III**. We start with font basics in Chapter 5 (shape, size, and so on) and continue with font encoding in Chapter 6: OT1, T1, and UTF-8. We show the usefulness of Postscript fonts such as the Times font, Math Time Pro 2, and Lucida Bright.

Part IV introduces *typing math* in \LaTeX .

Chapter 7 discusses inline formulas in detail, including basic constructs, delimiters, operators, math accents, and horizontally stretchable lines. The chapter concludes with a *Formula Gallery*.

Math symbols are covered in the first few sections in **Chapter 8**: how to space them, how to build new ones. We also introduce the STIX math symbols, some 2,000 of them, and look at the closely related subjects of math alphabets and fonts. Finally, we discuss tagging and grouping equations.

\LaTeX knows a lot about typesetting an “inline formula”, but not much about how to display a “multiline formula”. **Chapter 9** presents the numerous tools \LaTeX offers to help you do that. We start with a *Visual Guide* to help you get oriented. Then we discuss in detail the multitude of displaced math environments: `gather`, `multline`, `matrix`, ...

Part V discusses the parts of a \LaTeX document. In **Chapter 10**, you learn about the *structure* of a \LaTeX document. The most important topics are *sectioning* and *cross-referencing*. In **Chapter 11**, we discuss the `amsart` *document class* for articles. In particular, I present the title page information. `art2.tex` is featured in Chapter 11; it is a sample article for `amsart`, more advanced than the `art1.tex` typeset on the second page of Chapter 1. You can learn a lot about \LaTeX just by reading the source files one paragraph at a time and seeing how that paragraph is typeset. We conclude this chapter with a brief description of the AMS distribution, the packages and document classes, of which `amsart` is a part.

In **Part VI**, we start with **Chapter 12**, discussing *hyperlinks* and the `hyperref` package for PDF files. This prepares you for (slide) *presentations*, which are PDF files with hyperlinks. In **Chapter 13** we utilize the `beamer` *package* for making \LaTeX presentations and in **Chapter 14**, Michael Doob introduces its sister package `TikZ` for illustrations.

Part VII (Chapter 15) discusses techniques to *customize* \LaTeX : custom commands and environments created by users and command files. We present a sample command file, `newlattice.sty`, and a version of the second sample article utilizing this. You learn how parameters that affect \LaTeX ’s behavior are stored in counters and length commands, and how to change them. Also, how to design your own custom lists. A final section discusses the manifold pitfalls of customization.

In **Part VIII (Chapters 16–18)**, we discuss the special needs of longer documents. The application `BIBTeX` makes compiling *large numbered bibliographies* much

²For the mathematicians, physicists, computer scientists, statisticians, biologists, chemists ...

easier but it only provides numbered lists of references. **Chapter 17** discusses some more recent developments to do a variety of citation styles. the “Author-Year” system of citations used in humanities and social sciences.

The application *MakeIndex* in **Chapter 18** helps us compile *indexes*.

L^AT_EX provides the book and the *amsbook* document classes. They serve as foundations for well-designed books. We discuss these in **Chapter 19**. Better quality books use document classes designed by professionals.

You will probably find yourself referring to **Appendices A** and **B** time and again. They contain the *math and text symbol tables*. You can also find them as PDF files in the `samples` folder.

As I pointed it out, in the Preface to the Sixth Edition, I was greatly helped by ChatGPT in writing this edition.

While working on this book, I discovered unexpected assistance in ChatGPT. Frequently, I turned to ChatGPT for LaTeX commands, seeking definitions and examples. To provide the readers with the same help from ChatGPT, I added Appendix C, ChatGPT 101 for L^AT_EX users.

The text is supplemented by a collection of sample files, which will save you a good deal of typing. These are available for downloading; see Section 1.1.2. You also get PDF files of Mission Impossible, the symbol tables, and the STIX symbols.

Mission statement

This book is a guide for typesetting mathematical documents within the constraints imposed by L^AT_EX, an elaborate system with hundreds of rules. L^AT_EX allows you to perform almost any mathematical typesetting task through the appropriate application of its rules. You can customize L^AT_EX by introducing custom commands and environments and by changing L^AT_EX parameters. You can also extend L^AT_EX by invoking packages that accomplish special tasks.

It is *not my goal*:

- to survey the hundreds of L^AT_EX packages you can utilize to enhance L^AT_EX;
- to teach you how to write T_EX code to create your own packages;
- to discuss how to design beautiful documents by writing document classes.

The definitive book on the first topic is LC3. The second and third topics still await authoritative treatment.

Conventions

To make this book easy to read, I use some simple conventions:

- Explanatory text is set in this typeface: Times.

- Computer Modern typewriter is used to show what you should type, as well as messages from LaTeX. All the characters in this typeface have the same width, making the text easy to recognize.
- I also use Computer Modern typewriter to indicate:
 - commands (`\newpage`),
 - environments (`align`),
 - documents (`art1.tex`),
 - document classes (`amsart`),
 - document class options (`draft`),
 - folders or directories (`work`),
 - the names of *packages*—extensions of L^AT_EX (`verbatim`).
- When I show you how something looks when typeset, I use Computer Modern, T_EX’s standard typeface.

┌

I think you find this typeface sufficiently different from the other typefaces I have used. The strokes are much lighter so that you should not have much difficulty recognizing typeset L^AT_EX material. When the typeset material is a separate paragraph or paragraphs, corner brackets in the margin set it off from the rest of the text—unless it is a displayed formula.

└

- For explanations in the text, such as

Compare `iff` with `iff`, typed as `iff` and `if{f}`, respectively.

the same typeface is used. Because they are not set off spatially, it may be a little more difficult to see that `iff` is set in Computer Modern roman (in Times, it looks like this: `iff`), whereas `iff` is set in the Computer Modern typewriter typeface. Compare: `iff`, `iff`, `iff`, and a larger version: `iff`, `iff`, `iff`.

- I usually introduce commands by giving concrete examples, such as

`\[22pt]`

However, it is sometimes necessary to define the syntax of a command more formally. For instance, the formal syntax for the preceding example is as follows.

`\[length]`

where *length*, typeset in Computer Modern typewriter italic font, represents the value you have to supply.

Good luck and have fun.

A handwritten signature in black ink that reads "George Gratzer". The script is fluid and cursive, with the first name "George" and last name "Gratzer" clearly distinguishable.

E-mail: gratzer@mac.com

PART I

Mission Impossible



Short course

It happens to most of us. We live a happy life without \LaTeX and then, all of a sudden, we have to do something urgent that requires it.

If you are a student, maybe your professor turned to you and said “I need the solutions to these exercises typed up and distributed to the class by tomorrow” and the solutions are chock-full of formulas, difficult to do in Word.

Or you are a researcher whose documents have always been typed up by a secretary. You have to attend a conference and give a presentation. Your secretary is on vacation.

In my case, it was a letter (this was before e-mail) from the American Mathematical Society, in which they informed me that my paper, written in Word, was accepted for publication. The AMS will publish the paper in nine months. However, a \LaTeX version would be published in three months! So I had to learn \LaTeX in a hurry.

The mission, should you choose to accept it, is to get started really fast in \LaTeX . Our goal is to produce in \LaTeX the little article printed on the next page.

But we start with text into \LaTeX .

Relax, this chapter will not self-destruct in twenty seconds.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-55281-6_1

A TECHNICAL RESULT FOR CONGRUENCES OF FINITE LATTICES

G. GRÄTZER

ABSTRACT. We present a technical result for congruences on finite lattices.

1. INTRODUCTION

In some recent research, G. Czédli and I, see [1] and [2], spent quite an effort in proving that some equivalence relations on a planar semimodular lattice are congruences. The number of cases we had to consider was dramatically cut by the following result.

Theorem 1. *Let L be a finite lattice. Let δ be an equivalence relation on L with intervals as equivalence classes. Then δ is a congruence relation iff the following condition and its dual hold:*

(C₊) *If x is covered by y , $z \in L$ and $x \equiv y \pmod{\delta}$, then $z \equiv y + z \pmod{\delta}$.*

2. THE PROOF

We prove the join-substitution property: if $x \leq y$ and $x \equiv y \pmod{\delta}$, then

$$(1) \quad x + z \equiv y + z \pmod{\delta}.$$

Let $U = [x, y + z]$. We induct on $\text{length } U$, the length of U .

Let $I = [y_1, y + z]$ and $J = [z_1, y + z]$. Then $\text{length } I$ and $\text{length } J < \text{length } U$. Hence, the induction hypothesis applies to I and $\delta|_I$, and we obtain that $w \equiv y + w \pmod{\delta}$. By the transitivity of δ , we conclude that

$$(2) \quad z_1 \equiv y + w \pmod{\delta}.$$

Therefore, applying the induction hypothesis to J and $\delta|_J$, we conclude (1).

REFERENCES

- [1] G. Czédli, *Patch extensions and trajectory colorings of slim rectangular lattices*. Algebra Universalis **88** (2013), 255–280.
- [2] G. Grätzer, *Congruences of fork extensions of lattices*. Acta Sci. Math. (Szeged), **57** (2014), 417–434.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF MANITOBA, WINNIPEG, MB R3T 2N2, CANADA
 Email address, G. Grätzer: gratzer@mac.com
 URL, G. Grätzer: <http://tinyurl.com/gratzerhomepage>

Date: March 21, 2024.

2020 Mathematics Subject Classification. Primary: 06B10.

Key words and phrases. finite lattice, congruence.

1.1 Getting started

1.1.1 Your L^AT_EX

Are you sitting in front of your computer that has a L^AT_EX implementation? If you use a UNIX computer, you surely are. If you are in front of a Windows computer or a Mac, point your Internet browser at tug.org:

<https://tug.org>

Choose to download MikTeX for a Windows computer and MacTeX for a Mac. Follow the easy instructions, be patient, and you are done.

Even better, find a friend who can help.

1.1.2 Sample files

We work with many sample documents. The supplementary material is included per chapter in electronic form.

1. Create a folder, `samples`, on your computer to store the downloaded sample files.
2. Create also a second folder, called `work`, where you will keep your working files.
3. Copy the documents from the `samples` to the `work` folder as needed.
4. One of the sample files is `sample.cls`. Make sure it is in the `work` folder when you typeset a sample document.

In this book, the `samples` and `work` folders refer to the folders you created.

1.1.3 Editing cycle

Watch a friend type a document in L^AT_EX and learn the basic steps.

1. A text editor is used to create a L^AT_EX source file. A source file might look like this:

```
\documentclass{amsart}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\begin{document}
Then  $\delta$  is a congruence relation. I can type formulas!
\end{document}
```

Note that the source file is different from a typical word processor file. All characters are displayed in the same font and size.

2. Your friend “typesets” the source file (tells the application to produce a typeset version) and views the result on the monitor:

┌
└

Then δ is a congruence relation. I can type formulas!

3. *The editing cycle continues.* Your friend goes back and forth between the source file and the typeset version, making changes and observing the results of these changes.
4. *The file is viewed/printed.* View the typeset version as a pdf file or print it to get a paper version.

If \LaTeX finds a mistake when typesetting the source file, it records this in the *log file*. The *log window* (some call it *console*) displays a shorter version.

Various \LaTeX implementations have different names each for the source file, the text editor, the typeset file, the typeset window, the log file, and the log window. Become familiar with these names used by your system, so you can follow along with our discussions.

1.1.4 Typing the source file

A source file is made up of *text*, *formulas*, and *instructions (commands)* to \LaTeX . For instance, consider the following variant of the first sentence of this paragraph:

A source file is made up of text, formulas (e.g., $\sqrt{5}$), and *instructions to* \LaTeX .

This typesets as

┌
└

A source file is made up of text, formulas (e.g., $\sqrt{5}$), and *instructions to* \LaTeX .

In this sentence, the first part

A source file is made up of text, formulas (e.g.,
is text. Then

$\sqrt{5}$

is a formula, while

), and

is text again. Finally, we have

instructions to \LaTeX .

The instruction \emph is a *command with an argument*, while the instruction \LaTeX is a *command without an argument*. Commands, as a rule, start with a backslash (\backslash) and tell \LaTeX to do something special. In this case, the command \emph emphasizes its *argument* (the text between the braces). Another kind of instruction to \LaTeX is called an *environment*. For instance, the commands

```
\begin{center}
\end{center}
```

enclose a `center` environment; the *contents* (the text typed between these two commands) are centered when typeset.

In practice, text, formulas, and instructions (commands) are mixed. For example,

My first integral: $\int \zeta^2(x) dx$.

is a mixture of all three; it typesets as

My first integral: $\int \zeta^2(x) dx$.

Creating a document in \LaTeX requires that we type in the source file. So we start with the keyboard, proceed to type a short note, and learn some simple rules for typing text in \LaTeX .

1.2 The keyboard

The following keys are used to type the source file:

```
a-z    A-Z    0-9
+ = * / ( ) [ ]
```

You can also use the following punctuation marks:

```
, ; . ? ! : ' ' -
```

and the space bar, the Tab key, and the Return (or Enter) key.

Finally, there are thirteen special keys that are mostly used in \LaTeX commands:

```
# $ % & ~ _ ^ \ { } @ " |
```

If you need to have these characters typeset in your document, there are commands to produce them. For instance, the dollar sign, $\$$ is typed as `\$`, the underscore, $_$, is typed as `_`, and the percent sign, $\%$, is typed as `\%`. Only `@` requires no special command, type `@` to print `@`; see Section 3.1.2 and Appendix B.4.

There are also commands to produce composite characters, such as accented characters, for example \ddot{a} , which is typed as `\{"a}`. \LaTeX prohibits the use of other keys on your keyboard unless you have special support for it. See the text accent tables in Section 3.4.7 and Appendix B.2. If you want to use accented characters directly in your source file, then you must use the `inputenc` package.



Tip The text accent tables look formidable. Don't even dream of memorizing them. You will need to know very few accents. When you need a text accent, look it up. I recall

only one: `\{a}` (LOL). If you use a name with accented characters, figure out once how to type it, and then any time you need it, you can just copy and paste (chances are that the accented name is in your list of references).

1.3 Your first text note

We start our discussion on how to type a note in \LaTeX with a simple example. Suppose you want to use \LaTeX to produce the following:

It is of some concern to me that the terminology used in multi-section math courses is not uniform.

In several sections of the course on matrix theory, the term “hamiltonian-reduced” is used. I, personally, would rather call these “hyper-simple”. I invite others to comment on this problem.

To produce this typeset document, create a new text file, name it `textnote1.tex`, in the `work` folder, which you have created for this purpose, and make sure this folder also contains a copy of `sample.cls`, as discussed in Section 1.1.2. You can either copy the existing sample file `textnote1.tex` from your `samples` folder to the `work` folder, or create a new file with this name and type in the following, including the spacing and linebreaks shown, but not the line numbers:

```
1  % Sample file: textnote1.tex
2  \documentclass{sample}
3
4  \begin{document}
5  It is of some concern to me   that
6  the terminology used in   multi-section
7  math courses is not uniform.
8
9  In several sections of the course on
10 matrix theory, the term
11   ‘‘hamiltonian-reduced’’ is used.
12   I, personally, would rather call these
13   ‘‘hyper-simple’’. I invite others
14   to comment on this problem.
15 \end{document}
```

The first line of `textnote1.tex` starts with `%`. Such lines are called *comments* and are ignored by \LaTeX . Commenting is very useful. For example, if you want to add some notes to your source file and you do not want those notes to appear in the typeset version of your document, begin those lines with a `%`. You can also comment out part of a line:

simply put, we believe % actually, it's not so simple

Everything on the line after the % character is ignored by L^AT_EX.

Line 2 specifies our *document class*, `sample` (the special document class we provided for the sample documents), which controls how the document is formatted.

The text (or body) of the note is typed within the document environment, that is, between the lines `\begin{document}` and `\end{document}`.

Now typeset `textnote1.tex`. You should get the typeset document as shown at the beginning of this section. As you can see from this example, L^AT_EX is different from a word processor. It disregards the way you input and position the text, and follows only the formatting instructions given by the document class and your markup commands. L^AT_EX notices when you put a blank space in the text, but it ignores *how many blank spaces* have been typed. L^AT_EX does not distinguish between a blank space (hitting the space bar), a tab (hitting the Tab key), and a *single* carriage return (hitting Return once). However, hitting Return twice gives a blank line; *one or more* blank lines mark the end of a paragraph. There is also an explicit command for a *new paragraph*: `\par`.

L^AT_EX, by default, fully justifies text by placing a flexible amount of space between words—the *interword space*—and a somewhat larger space between sentences—the *intersentence space*. If you have to force an interword space, you can use the `_` command (L^AT_EX books use the symbol $_$ for blank spaces). The `~` (tilde) command also forces an interword space, but with a difference: it keeps the words on the same line. This command produces a *tie* or *nonbreakable space*.

Note that on lines 11 and 13, the left double quotes are typed as two left single quotes and the right double quote is typed as two right single quotes, apostrophes.

We numbered the lines of the source file for easy reference. Sometimes you may want the same for the typeset file. This is really easy. Just add the two lines

```
\usepackage{lineno}
\linenumbers
```

after the `\documentclass` line and you get:

1 It is of some concern to me that the terminology used in multi-section math
 2 courses is not uniform.
 3 In several sections of the course on matrix theory, the term “hamiltonian-
 4 reduced” is used. I, personally, would rather call these “hyper-simple”. I invite
 5 others to comment on this problem.

Next, we produce the following note:

January 5, 2024

From the desk of George Grätzer

February 7–21 *please* use my temporary e-mail address:

George_Gratzer@yahoo.com

Type the source file, without the line numbers. Save it in your work folder as `textnote2.tex` (`textnote2.tex` can also be found in the `samples` folder):

```
1 % Sample file: textnote2.tex
2 \documentclass{sample}
3
4 \begin{document}
5 \begin{flushright}
6   \today
7 \end{flushright}
8 \textbf{From the desk of George Gr\"atzer}
9
10 February 7--21 \emph{please} use my
11 temporary e-mail address:
12 \begin{center}
13   \texttt{George\_Gratzer@yahoo.com}
14 \end{center}
15 \end{document}
```

This note introduces several additional text features of \LaTeX :

- The `\today` command (in line 6) to display the date on which the document is typeset, so you will see a date different from the date shown above in your own typeset document (see also Section 3.4.8).
- The environments to *right align* (lines 5–7) and to *center* (lines 12–14) text.

- The commands to change the text style, including the `\emph` command (line 10 in the second paragraph) to *emphasize* text, the `\textbf` command (line 8 in the first paragraph) for **bold** text (text bold font), and the `\texttt` command (line 13 in the second paragraph) to produce typewriter style text. These are *commands with arguments* on which the command acts.
- The form of the \LaTeX commands. As we have noted already, almost all \LaTeX *commands* start with a backslash (`\`) followed by the *command name*. For instance, `\textbf` is a valid command and `textbf` is the command name. The command name is terminated by the first *non-alphabetic character*, that is, by any character other than a–z or A–Z.



Tip `textnote2.tex` is a valid file name but `textbf1` is not a command name. `\textbf1` typesets as 1. Let's look at this a bit more closely. `\textbf` is a valid command. If a command needs an argument and it is not followed by braces, then it takes the next character as its argument. So `\textbf1` is the command `\textbf` with the argument 1; it typesets as 1.

- The multiple role of hyphens: Double hyphens are used for number ranges. For example, 7--21 (in line 10) typesets as 7–21. The punctuation mark – is called an *en dash*. Use triple hyphens for the *em dash* punctuation mark—such as the one in this sentence.
- Special rules for *special characters* (see Section 1.2), for *accented letters*, and for some *European characters*. For instance, the accented letter ä is typed as `\{"a}`. But I confess, I always type my name as `Gr\atzer`, without the braces. (Even \LaTeX is changing. If you use “T1 encoding”, then the rules are different. This is discussed in depth in Chapter 6, for the time being, ignore it.)

See Section 3.4 for more detail. In Appendix B, all the text symbols are organized into tables. We also have the `SymbolTables.pdf` in the `samples` folder.



Tip Keep `SymbolTables.pdf` handy on your computer!

1.4 Lines too wide

\LaTeX reads the text in the source file one line at a time and typesets the entire paragraph when the end of a paragraph is reached. Occasionally, \LaTeX gets into trouble when trying to split the paragraph into typeset lines. To illustrate this situation, modify `textnote1.tex`. In the second sentence, replace `term` by `strange term`. Now save

this modified file in your work folder using the name `textnote1bad.tex` (or copy the file from the `samples` folder).

Typesetting `textnote1bad.tex`, you obtain the following:

┌

It is of some concern to me that the terminology used in multi-section math courses is not uniform.

In several sections of the course on matrix theory, the strange term “hamiltonian-reduced” is used. I, personally, would rather call these “hyper-simple”. I invite others to comment on this problem.

└

The first line of paragraph two is too wide. In the log window, \LaTeX displays the following messages:

```
Overfull \hbox (15.38948pt
too wide) in paragraph at lines 9--15 []\OT1/cmr/m/n/10 In sev-eral
sec-tions of the course on ma-trix the-ory, the strange term
‘‘hamiltonian-
```

It informs you that the typeset version of this paragraph has a line that is 15.38948 points too wide. \LaTeX uses *points* (pt) to measure distances; there are about 72 points in one inch. (See also the discussion in Section 3.6.1.) Then it identifies the source of the problem: \LaTeX did not properly hyphenate the word `hamiltonian-reduced` because it by default hyphenates a hyphenated word *only at the hyphen*. What to do, when a line is too long?



Tip Your first line of defense: reword the offending line. Write

```
The strange term ‘‘hamiltonian-reduced’’ is used
in several sections of the course on matrix theory.
```

and the problem goes away.

Your second line of defense: insert one or more *optional hyphen commands* (`\-`), which tell \LaTeX where it can hyphenate the word. Write:

```
hamil\-\tonian-reduced
```

1.5 A note with formulas

In addition to the regular text keys and the 13 special keys discussed in Section 1.2, two more keys are used to type formulas: `<` and `>`. The formula $2 < |x| > y$ (typed as `$2 < |x| > y$`) uses both. Note that such a formula, called *inline*, is enclosed by a pair of `$` symbols.

We begin typesetting formulas with the following note:

┌ In first-year calculus, we define intervals such as (u, v) and (u, ∞) . Such an interval is a *neighborhood* of a if a is in the interval. Students should realize that ∞ is only a symbol, not a number. This is important since we soon introduce concepts such as $\lim_{x \rightarrow \infty} f(x)$.

When we introduce the derivative

$$\lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a},$$

└ we assume that the function is defined and continuous in a neighborhood of a .

To create the source file for this mixed text and formula note, create a new document with your text editor. Name it `formulanote.tex`, place it in the work folder, and type the following, without the line numbers (or simply copy `formulanote.tex` from the samples folder):

```

1  % Sample file: formulanote.tex
2  \documentclass{sample}
3
4  \begin{document}
5  In first-year calculus, we define intervals such
6  as  $(u, v)$  and  $(u, \infty)$ . Such an interval
7  is a \emph{neighborhood} of  $a$ 
8  if  $a$  is in the interval. Students should
9  realize that  $\infty$  is only a
10 symbol, not a number. This is important since
11 we soon introduce concepts
12 such as  $\lim_{x \rightarrow \infty} f(x)$ .
13
14 When we introduce the derivative
15 \[
16     \lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a},
17 \]
18 we assume that the function is defined and
19 continuous in a neighborhood of  $a$ .
20 \end{document}
```

This note introduces several basic concepts of formulas in L^AT_EX:

- There are two kinds of math formulas and environments in `formulanote.tex`:
 - *Inline* formulas; they open and close with `$` or open with `\(` and close with `\)`—is anybody using these?

- *Displayed* math environments; they open with `\[` and close with `\]`. (We will introduce many other displayed math environments in Section 1.7 and Chapter 9.)
- L^AT_EX uses its own spacing rules within math environments, and completely ignores the white spaces you type, with two exceptions:
 - Spaces that terminate commands. So in `$(\infty a$` the space is not ignored; `$(\infty a$` produces an error.
 - Spaces in the arguments of commands that temporarily revert to regular text. `\text` is such a command; see Sections 1.6 and 7.4.6.

The white space that you add when typing formulas is important only for the readability of the source file.

- A math symbol is invoked by a command. For example, the command for ∞ is `\infty` and the command for \rightarrow is `\to`. The math symbols are organized into tables in Appendix A; see also `SymbolTables.pdf` in the `samples` folder.
- Some commands, such as `\sqrt`, need *arguments*, which are enclosed by `{` and `}`. To typeset $\sqrt{5}$, type `$(\sqrt{5}$`, where `\sqrt` is the command and 5 is the argument. Some commands need more than one argument. To get

$$\frac{3+x}{5}$$

type

```
\[
  \frac{3+x}{5}
\]
```

where `\frac` is the command, `3+x` and `5` are the arguments.

- Do not leave a blank line before a displayed formula!



Tip Keep in mind that many spaces equal one space in text, whereas your spacing is ignored in formulas, unless the space terminates a command.

1.6 The building blocks of a formula

A formula (inline or displayed) is built from various types of components. We group them as follows:

- arithmetic,
- binomial coefficients,

congruences,
 delimiters,
 ellipses,
 integrals,
 math accents,
 matrices,
 operators,
 roots,
 text.

In this section, I describe all of these groups, and provide examples illustrating their use. Read carefully the groups you need!

Arithmetic We type the arithmetic operations $a + b$, $a - b$, $-a$, a/b , and ab in the natural way: `$a + b$`, `$a - b$`, `$-a$`, `a / b`, and `$a b$` (the spaces are typed only for readability).

If you wish to use \cdot or \times for multiplication, as in $a \cdot b$ or $a \times b$, use `\cdot` or `\times`, respectively. The formulas $a \cdot b$ and $a \times b$ are typed as `$a \cdot b$` and `$a \times b$`.

Displayed fractions, such as

$$\frac{1 + 2x}{x + y + xy}$$

are typed with `\frac`:

```
\[
  \frac{1 + 2x}{x + y + xy}
\]
```

Subscripts and superscripts Subscripts are typed with `_` and superscripts with `^` (caret). Subscripts and superscripts should be enclosed in braces, that is, typed between `{` and `}`. To get a_1 , type `a_{1}`. Omitting the braces in this example causes no harm, but to get a_{10} , you *must* type `a_{10}`. Indeed, `a_10` is typeset as a_10 .

There is one symbol, the prime (`'`), that is automatically superscripted in a formula. To get $f'(x)$, just type `$f'(x)$`. See Section 7.4.1 for more detail.

Binomial coefficients Binomial coefficients are typeset with the `\binom` command. `\binom{a}{b + c}` produces inline: $\binom{a}{b+c}$, whereas

$$\binom{a}{b+c}$$

is the displayed version. See Section 7.4.2 for more detail.

Congruences The two most important forms are

$$\begin{array}{ll} a \equiv v \pmod{\theta} & \text{typed as } \$a \equiv v \pmod{\theta}$ \\ a \equiv v (\theta) & \text{typed as } \$a \equiv v \pod{\theta}$ \end{array}$$

See Section 7.6.2 for more detail.

Delimiters Parentheses and square brackets are examples of delimiters. These are used to delimit some subformulas, as in $[(a*b)+(c*d)]^2$, which typesets inline as $[(a * b) + (c * d)]^2$. L^AT_EX can be instructed to expand them vertically to enclose a formula such as

$$\left(\frac{1+x}{2+y^2} \right)^2$$

which is typed as

$$\left[\left(\frac{1 + x}{2 + y^2} \right)^2 \right]$$

The `\left(` and `\right)` commands tell L^AT_EX to size the parentheses correctly, relative to the size of the formula inside the parentheses; sometimes the result is pleasing, sometimes not. We dedicate Section 7.5 to this topic.

Ellipses In a formula, the ellipsis is printed either as *low* (or *on-the-line*) *dots*:

$$F(x_1, \dots, x_n) \text{ is typed as } \$F(x_{\{1\}}, \dots, x_{\{n\}})$$$

or as *centered dots*:

$$x_1 + \cdots + x_n \text{ is typed as}$$

$$\$x_{\{1\}} + \dots + x_{\{n\}}$$$

Use `\ldots` and `\cdots` if `\dots` does not work as expected. See Section 7.4.3 for more detail.

Integrals The command for an integral is `\int`. The lower limit is specified as a subscript and the upper limit is specified as a superscript. For example, the formula $\int_0^\pi \sin x \, dx = 2$ is typed as

$$\$ \int_0^{\pi} \sin x \, dx = 2 $$$

where `\,` is a spacing command.

The formula looks bad without the spacing command: $\int_0^\pi \sin x dx = 2$. See Section 7.4.4 for more complicated integrals.

Math accents The four most frequently used math accents are:

\bar{a} typed as `\bar{a}` \hat{a} typed as `\hat{a}`
 \tilde{a} typed as `\tilde{a}` \vec{a} typed as `\vec{a}`

See Section 7.7 for more detail. See Sections 7.7 and A.7 for complete lists.

Matrices Type the matrix

$$\begin{matrix} a + b + c & uv & x - y & 27 \\ a + b & u + v & z & 134 \end{matrix}$$

with the `\matrix` command

```
\[
  \begin{matrix}
    a + b + c & & uv & & x - y & & 27 \\
    a + b & & u + v & & z & & 134
  \end{matrix}
\]
```

The `matrix` environment separates adjacent matrix elements within a row with ampersands. Rows are *separated* by new line commands, `\\`.



Tip Do not end the last row with a new line command.

The `matrix` environment has to appear within a formula, as a rule, in a displayed formula. It can be used in the `align` environment which is discussed in Section 1.7.3 and Appendix 9.5.

The `matrix` environment does not provide delimiters. Several variants do, including `pmatrix` and `vmatrix`. For example, the equation :

$$\mathbf{A} = \begin{pmatrix} a + b + c & uv \\ a + b & u + v \end{pmatrix} \begin{vmatrix} 30 & 7 \\ 3 & 17 \end{vmatrix}$$

is typed as follows (for the `\mathbf` command; see Section 8.4.1):

```
\[
  \mathbf{A} =
  \begin{pmatrix}
    a + b + c & uv \\
    a + b & u + v
  \end{pmatrix}
\]
```

```

\begin{vmatrix}
30 & 7 \\
3 & 17
\end{vmatrix}

```

As you can see, `pmatrix` typesets as a matrix between a pair of `\left(` and `\right)` commands, while `vmatrix` typesets as a matrix between a pair of `\left|` and `\right|` commands. There is also `bmatrix` for square brackets.

See Section 9.7.1 for a listing of all the matrix variants and Section 7.5 and Appendix A.5 for lists of delimiters.

Operators To typeset the sine function, $\sin x$, type `\sin x`. Note that `\sin x` would be typeset as $\sin x$ —how awful. L^AT_EX calls `\sin` an *operator*. Section 7.6 and Appendix A.6 list a number of operators. Some are just like `\sin`. Others produce a more complex display with limit, for example,

$$\lim_{x \rightarrow 0} f(x) = 0$$

is typed as

```

\[
\lim_{x \to 0} f(x) = 0
\]

```

See Section 7.6 for more detail.

Large operators The command for *sum* is `\sum` and for *product* is `\prod`. The following two examples (see Section 8.1.3 for the `\quad` command):

$$\sum_{i=1}^n x_i^2 \quad \prod_{i=1}^n x_i^2$$

are typed as

```

\[
\sum_{i=1}^n x_i^2 \quad \prod_{i=1}^n x_i^2
\]

```

Sum and product are examples of *large operators*. They are typeset larger in displayed math than in an inline formula. They are listed in Section 7.6.3 and Appendix A.6.1. See Section 7.6.3 for more detail.

Roots `\sqrt` produces a square root. `\sqrt{a + 2b}` typesets as $\sqrt{a + 2b}$. The n -th root, $\sqrt[n]{5}$, requires the use of an *optional argument*, which is specified in brackets: `\sqrt[n]{5}`. See Section 7.4.5.

Text You can include text in a formula with a `\text` command. For instance,

$$a = b, \quad \text{by assumption,}$$

is typed as

```
\[
  a = b, \text{\quad by assumption},
\]
```

where `\quad` is a spacing command.

See Section 7.4.6 for more detail.

1.7 Displayed formulas

1.7.1 Equations

The `equation` environment creates a displayed formula and automatically generates an equation number. The equation

$$(1) \qquad \int_0^\pi \sin x \, dx = 2$$

is typed as

```
\begin{equation}\label{E:firstIntegral}
  \int_0^\pi \sin x \, dx = 2
\end{equation}
```

The equation number, which is automatically generated, depends on how many numbered displayed formulas occur before the given equation. You can choose to have equations numbered within each section—(1.1), (1.2), ..., in Section 1; (2.1), (2.2), ..., in Section 2; and so on—by including, in the preamble (see Sections 1.8 and 7.3), the command

```
\numberwithin{equation}{section}
```

You can choose to have the equation numbers on the right, see the `reqno` option of the `amsart` document class in Section 11.5.

The `equation*` environment accomplishes the same as the `equation` environment, without equation numbers. The commands `\[` and `\]` are equivalent to

```
\begin{equation*} \qquad \end{equation*}
```

You may want to use this environment for the ease of deleting the `*`-s if you wish.

1.7.2 Symbolic referencing

To reference a formula without having to remember its number—which can change when you edit your document—give the equation a symbolic label by using the `\label` command, then refer to the equation in your document by using the symbolic label, the argument of the `\label` command. In this example, I have called the first equation `firstIntegral`, and used the convention that the label of an equation starts with `E:`, so that the complete `\label` command is `\label{E:firstIntegral}`.

The number of this formula is referenced with the `\ref` command. Its page is referenced using the `\pageref` command. For example, to get

see (1) on page 19.

type (see Sections 1.3 and Section 3.4.3 for the nonbreaking ~)

`see~(\ref{E:firstIntegral}) on page~\pageref{E:firstIntegral}.`

The `\eqref` command provides the reference number in parentheses. So the last example could be typed

`see~\eqref{E:firstIntegral} on page~\pageref{E:firstIntegral}.`

The `\eqref` command is smart. Even if the equation number is referenced in emphasized or italicized text, the reference typesets upright (in roman type).

The main advantage of this cross-referencing system is that when you add, delete, or rearrange equations, \LaTeX automatically rennumbers the equations and adjusts the references that appear in your typeset document. For bibliographic references, \LaTeX uses the `\bibitem` command to define a bibliographic item and the `\cite` command to cite it.



Tip For renumbering to work, typeset **twice** after editing the file.



Tip It is a good idea to check the \LaTeX warnings periodically in the log file. If you forget to typeset the source file twice when necessary, \LaTeX issues a warning.

What happens if you misspell a reference, e.g., typing `\ref{E:FirstIntegral}` instead of `\ref{E:firstIntegral}`? \LaTeX typesets `??`. There are two warnings in the log file:

LaTeX Warning: Reference ‘E:FirstIntegral’ on page 39
undefined on input line 475.

for the typeset page and the other one close to the end:

LaTeX Warning: There were undefined references.

If the argument of `\cite` is misspelled, you get [?] and similar warnings.

Check the **Tip** on page 72.

Absolute referencing

Equations can also be *tagged* by attaching a name to the formula with the `\tag` command. The tag replaces the equation number.

For example,

$$(Int) \qquad \int_0^{\pi} \sin x \, dx = 2$$

is typed as

```
\begin{equation}
  \int_{0}^{\pi} \sin x \, dx = 2 \tag{Int}
\end{equation}
```

Tags are *absolute*. This equation will *always* be referred to as (Int). Equation numbers, on the other hand, are *relative*, they may change when the file is edited.

1.7.3 Aligned formulas

L^AT_EX has many ways to typeset multiline formulas. We discuss three constructs in this section: *simple alignment*, *annotated alignment*, and *cases*. For (so many) more constructs; see Chapter 9.

We use the `align` environment for simple and annotated alignment. Each line in the `align` environment is a separate equation, which L^AT_EX automatically numbers.

Simple alignment

Simple alignment is used to align two or more formulas. To obtain the formulas

$$\begin{aligned} (2) \qquad & r^2 = s^2 + t^2, \\ (3) \qquad & 2u + 1 = v + w^\alpha, \\ (4) \qquad & x = \frac{y + z}{\sqrt{s + 2u}}; \end{aligned}$$

type the following, using `\\` as a *line separator* and `&` as an *alignment point*:

```
r^{2}      &= s^{2} + t^{2},                \label{E:Pyth}\\
2u + 1     &= v + w^{\alpha},             \label{E:alpha}\\
x          &= \frac{y + z}{\sqrt{s + 2u}};   \label{E:frac}
```

Figure 1.1 may help visualize the placements of the ampersands.



Tip In this displayed formula, `\\` is a *line separator*, not a new line command. Do not place the command `\\` to terminate the last line!

r^2	<code>&= s^2 + t^2,</code>	<code>\label{E:Pyth}\\</code>
$2u + 1$	<code>&= v + w^{\alpha},</code>	<code>\label{E:alpha}\\</code>
x	<code>&= \frac{y + z}{\sqrt{s + 2u}};</code>	<code>\label{E:frac}</code>

alignment points
of formulas

(2)	$r^2 = s^2 + t^2,$
(3)	$2u + 1 = v + w^\alpha,$
(4)	$x = \frac{y + z}{\sqrt{s + 2u}};$

Figure 1.1: Simple alignment: source and typeset.

These formulas are numbered (2) – (4) because they are preceded by one numbered equation earlier in this section.

The `align` environment can also be used to break a long formula into two or more parts. Since numbering both lines in such a case would be undesirable, you can prevent the numbering of the second line by using the `\notag` command. For example,

$$\begin{aligned}
 (5) \quad h(x) &= \int \left(\frac{f(x) + g(x)}{1 + f^2(x)} + \frac{1 + f(x)g(x)}{\sqrt{1 - \sin x}} \right) dx \\
 &= \int \frac{1 + f(x)}{1 + g(x)} dx - 2 \tan^{-1}(x - 2)
 \end{aligned}$$

is typed as follows:

```

\begin{align}
h(x) &= \int \left( \frac{f(x) + g(x)}{1 + f^2(x)} \right. \\
&\quad \left. + \frac{1 + f(x)g(x)}{\sqrt{1 - \sin x}} \right) dx \label{E:longInt} \\
&= \int \frac{1 + f(x)}{1 + g(x)} dx \\
&\quad - 2 \tan^{-1}(x-2) \notag
\end{align}

```

The rules for simple alignment are easy to remember.



Rule: Simple alignments.

- Use the `align` environment.
 - *Separate* the lines with `\\`.
 - In each line, indicate the alignment point with `&`, one `&` per line. If the alignment point is adjacent to an `=`, `+`, and so on, place the `&` *before* to ensure proper spacing.
 - Place a `\notag` command in each line that you do not wish numbered.
 - If no line should be numbered, use the `align*` environment.
 - Place a `\label` command in each numbered line you want to reference with the commands `\ref`, `\eqref`, or `\pageref`.
-

Annotated alignment

Annotated alignment allows you to align formulas and their annotations, that is, explanatory text, separately:

$$\begin{array}{ll}
 (6) & x = x \wedge (y \vee z) & \text{(by distributivity)} \\
 & = (x \wedge y) \vee (x \wedge z) & \text{(by condition (M))} \\
 & = y \vee z
 \end{array}$$

This is typed as

```

\begin{align}
x &= x \wedge (y \vee z) \\
&\quad \&\text{(by distributivity)} \label{E:Align} \eqref{E:Align} \\
&= (x \wedge y) \vee (x \wedge z) \\
&\quad \&\text{(by condition (M))} \notag \\
&= y \vee z \notag
\end{align}

```

aligned formulas	annotation
$ \begin{array}{l} x \mid \&= x \wedge (y \vee z) \\ \mid \&= (x \wedge y) \vee (x \wedge z) \\ \mid \&= y \vee z. \end{array} $	$ \begin{array}{l} \mid \&\&\text{\texttt{(by distributivity)}} \\ \mid \&\&\text{\texttt{(by condition (M))}} \end{array} $
alignment points of formulas	alignment points of annotations

aligned formulas	annotation
$ \begin{array}{l} x \mid = x \wedge (y \vee z) \\ \mid = (x \wedge y) \vee (x \wedge z) \\ \mid = y \vee z. \end{array} $	$ \begin{array}{l} \mid (\text{by distributivity}) \\ \mid (\text{by condition (M)}) \end{array} $
alignment points of formulas	alignment points of annotations

Figure 1.2: Annotated alignment: source and typeset.

Figure 1.2 may help visualize the placements of the ampersands.



Rule: Annotated alignment.

The rules for annotated alignment are similar to the rules of simple alignment. In each line, in addition to the alignment point marked by $\&$, there is also a mark for the start of the annotation: $\&\&$.

1.7.4 Cases

The `cases` construct is a specialized matrix. It has to appear within a math environment such as the `equation` environment or the `align` environment. Here is a typical example:

$$f(x) = \begin{cases} -x^2, & \text{if } x < 0; \\ \alpha + x, & \text{if } 0 \leq x \leq 1; \\ x^2, & \text{otherwise.} \end{cases}$$

It is typed as follows:

```

\[
  f(x)=
  \begin{cases}
    -x^2, & \&\text{if } x < 0;\}\\
    \alpha + x, & \&\text{if } 0 \leq x \leq 1;\}\\
    x^2, & \&\text{otherwise.}
  \end{cases}
\]
```

The rules for using the `cases` environment are the same as for matrices. Separate the lines with `\\` and indicate the annotation with `&`.

1.8 The anatomy of a document

To begin, we use the sample document `art1.tex` (in the `samples` folder) to examine the anatomy of an document.

Every \LaTeX document has two parts, the preamble and the body. The *preamble* of a document is everything from the first line of the source file down to the line

```
\begin{document}
```

The *body* is the content of the document environment. For a schematic view of a document; see Figure 1.3.

The preamble contains instructions affecting the entire document. The *only* required command in the preamble is the `\documentclass` command. There are other commands (such as the `\usepackage` commands; see Section 10.2) that must be placed in the preamble if they are used, but such commands do not have to be present in every document.

Here is the preamble and top matter of `art1`:

```

%First article, art1.tex
\documentclass{amsart}
\usepackage{amssymb,latexsym}
\newtheorem{theorem}{Theorem}

\begin{document}
\title{A technical result\\ for congruences of finite lattices}
\author{G. Gr\"atzer}
\address{Department of Mathematics\\
  University of Manitoba\\
  Winnipeg, MB R3T 2N2\\
  Canada}
\email{G. Gr\"atzer}{gratzer@mac.com}
\urladdr{G. Gr\"atzer}{http://tinyurl.com/gratzerhomepage}
```

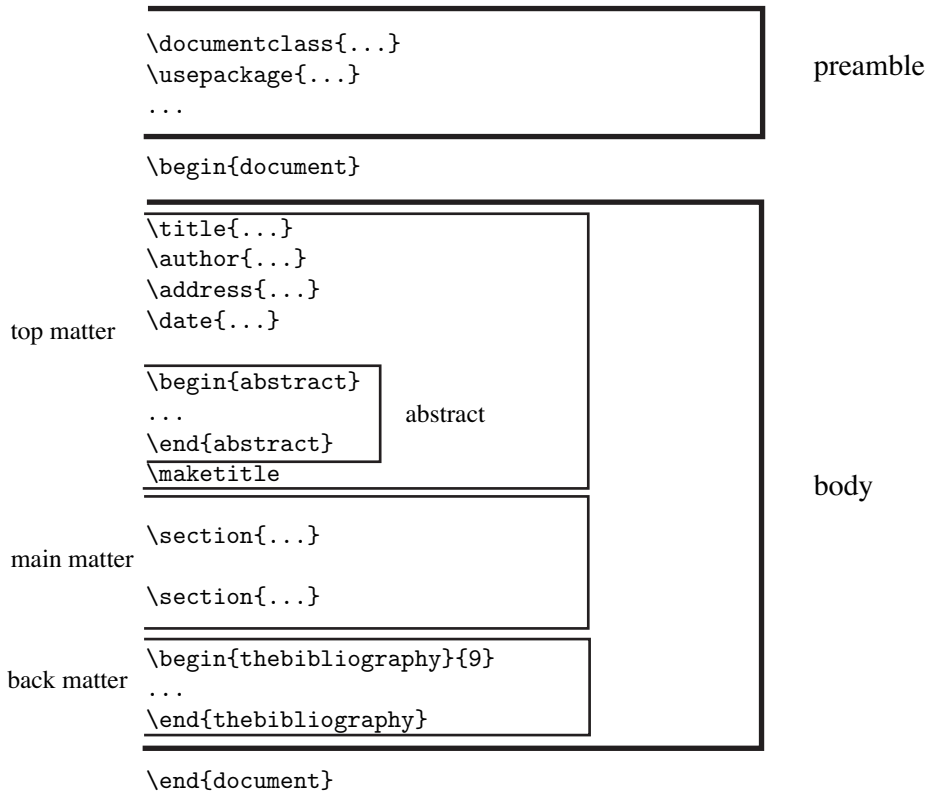


Figure 1.3: A schematic view of a document.

```
\date{March 21, 2024}
\subjclass[2020]{Primary: 06B10.}
\keywords{finite lattice, congruence.}
```

```
\maketitle
\begin{abstract}
We present a technical result for congruences on finite lattices.
\end{abstract}
```

You find the source file, `art1.tex`, in the `samples` folder and the typeset document on page 4.

To simplify the discussion in Part I, we discuss only one document class for articles: `amsart`. You may come across its predecessor, the `article` document class, which handles a limited set of commands for the preamble and the top matter and also displays them differently. We shall discuss in detail the `amsart` document class in Chapter 11. For the `article` document class; see Section 11.7.

1.9 Your own commands

Over time, \LaTeX can be adjusted to fit your needs. You add packages to enable \LaTeX to do new things (such as the `graphicx` package in the next section) and introduce your own commands to facilitate typing and make the source file more readable.

We can add two new commands (called custom commands; see Section 1.9) to the sample article `art1.tex`:

```
\newcommand{\pdelta}{\pmod{\delta}}
\DeclareMathOperator{\length}{length}
```

So instead of

```
$x \equiv y \pmod{\delta}$
```

we can type

```
$x \equiv y \pdelta$
```

and instead of `length\,,U,` we can type `$\length\ U$` (see Section 15.1.6). Notice how the spacing is now done by \LaTeX !

We dedicate Chapter 15 to \LaTeX customization.

1.10 Adding an illustration

“And what is the use of a book”, thought Alice, “without pictures or conversations?” I am not sure what to suggest about conversations, but illustrations (graphics) we can tackle with ease. Let us add an illustration, `covers.pdf` to `art1`. First, add

```
\usepackage{graphicx}
```

to the preamble as the fourth line of the document. This will enable \LaTeX to tackle illustrations. Secondly, add the following lines to `art1.tex`, say, as the second paragraph of the introduction, to get `art1i11.tex`:

```
\begin{figure}[hbt]
\centering{\includegraphics{covers}}
\caption{Theorem~\ref{T:technical} illustrated}\label{I:Theorem}
\end{figure}
```

We place the illustration `covers.pdf` in the same folder as `art1i11.tex`. That’s it. You find `covers.pdf` and `art1ind.tex` in the `samples` folder.



Tip Make sure that the `\label` command follows the `\caption` command! You may have a hard time explaining some troubles that develop otherwise.

See Section 10.4.3 for more information on illustrations.

Many people in my field use the vector graphics application Adobe Illustrator to produce the PDF files for illustrations. In Chapter 14, we discuss an alternative, TikZ, built for L^AT_EX. INKSCAPE is also an alternative, available for all platforms.

1.11 The anatomy of a presentation

Chances are, one of your first exposures to L^AT_EX was watching a *presentation*. The presenter used a pdf document produced by L^AT_EX and opened it with Adobe Reader. The presenter went from “slide” to “slide” by pressing the space bar. Figures 1.4 and Figure 1.5 show four slides of a presentation.

In L^AT_EX, you use a presentation package—really, a document class—to prepare the PDF file. We use Till Tantau’s BEAMER.

Here are the first few lines—the preamble and the Title slide—of the source file of our sample presentation, `firstpresentation.tex`; see it in the `samples` folder.

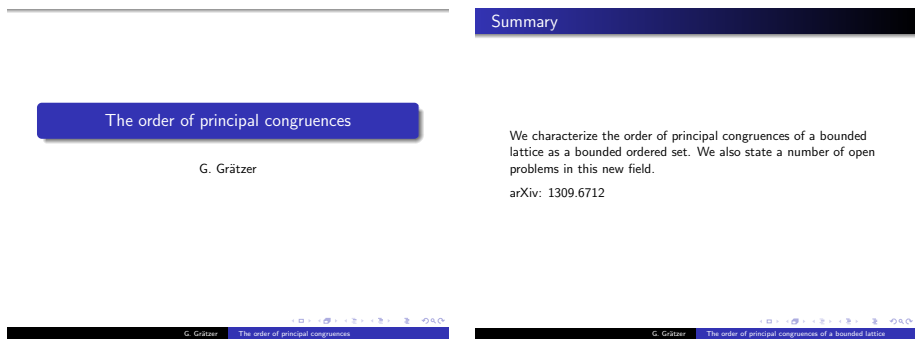


Figure 1.4: The Title slide (Slide 1) and Slide 2

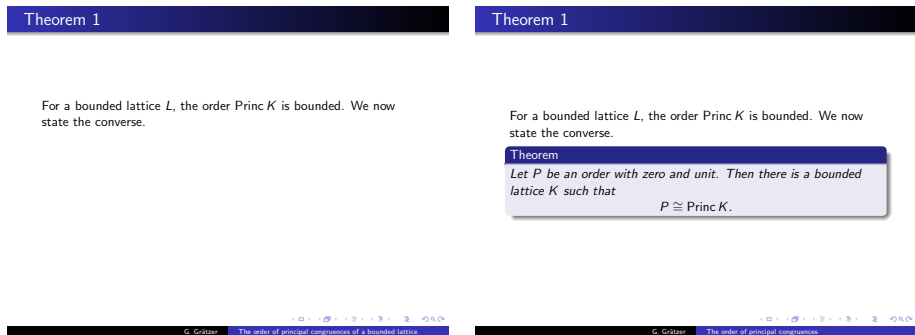


Figure 1.5: Slides 3 and 4

```

\documentclass[leqno]{beamer}
\usetheme{Warsaw}

\DeclareMathOperator{\Princ}{Princ}

\begin{document}
\title{The order of principal congruences}
\author{G. Gr"atzer}
\date{}
\maketitle

```

`\usetheme{Warsaw}` provides a flavor. It is followed by the Title slide, providing the title and the author.

The `\title` command may be longer—it may contain all the additional information you may want to display. Here is the `\title` command of another presentation:

```

\title[The order of principal congruences of a bounded lattice]
{The order of principal congruences\\
of a bounded lattice.\\
AMS Fall Southeastern Sectional Meeting\\
October 5-6, 2023}

```

Note that this `\title` has two parts. The first, in `[]`, is the *short title*, repeated in the bottom line on every slide. The second, in `{}`, is the *long title*, the title for the front page.

The rest of the presentation source file is divided into *frames* with the structure:

```

\begin{frame}
\frametitle{}
\end{frame}

```

Each frame produces a “slide” (or more). Here is the first frame:

```

\begin{frame}
\frametitle{Summary}
We characterize the order of principal congruences
of a bounded lattice
as a bounded ordered set.
We also state a number of open problems in this new field.
\medskip

arXiv:1309.6712
\end{frame}

```

The command `\frametitle` gives the slide its title: Summary, see Slide 2 in Figure 1.4. In the body of the frame, you type regular \LaTeX .

To produce Slides 3 and 4, it would be natural to try

```
\begin{frame}
\frametitle{Theorem 1}
For a bounded lattice  $L$ , the order  $\Princ K$  is bounded.
We now state the converse.
\end{frame}

\begin{frame}
\frametitle{Theorem 1}
For a bounded lattice  $L$ , the order  $\Princ K$  is bounded.
We now state the converse.
\begin{theorem}
Let  $P$  be an order with zero and unit.
Then there is a bounded lattice  $K$  such that
\[
P \cong \Princ K.
\]
If  $P$  is finite, we can construct  $K$  as a finite lattice.
\end{theorem}
\end{frame}
```

which produces the two frames of Figure 1.6.

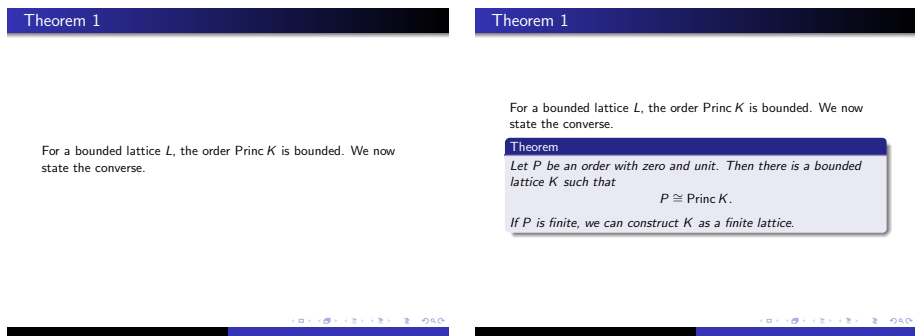


Figure 1.6: Slides 3 and 4, first try

This is really jarring to watch. The two lines of the new Slide 3 jump up more than two lines as they transition to Slide 4.

Here is how we produce Slides 3 and 4 of Figure 1.5:

```
\begin{frame}
\frametitle{Theorem 1}
```

```

For a bounded lattice  $L$ , the order  $\Princ K$  is bounded.
We now state the converse.
\pause
\begin{theorem}
Let  $P$  be an order with zero and unit.
Then there is a bounded lattice  $K$  such that
\[
P \cong \Princ K.
\]
If  $P$  is finite, we can construct  $K$  as a finite lattice.
\end{theorem}
\end{frame}
\end{document}

```

There is only one new command to learn: `\pause`; it produces **two** slides from this frame.

The `\pause` in this frame splits the contents of the frame into two parts. The first slide is typeset from the first part as if the second part was also present. The second slide is typeset from both parts. So the transition from the first slide to the second is smooth; see Figure 1.5.

You can have more than one `\pause` in a frame. Use `\pause` also to display a list one item at a time.

Chapter 13 discusses BEAMER in more detail.



And one more thing

A few more things . . .

If life was perfect, we would not need this chapter. You would write perfect \LaTeX , based on Chapter 1, no need to study how \LaTeX works, what error messages mean, and much more. But life is not perfect, you will make mistakes, and \LaTeX will send messages, some clear and some exceedingly mysterious.

In this chapter, we briefly explain how everything works, the structure of \LaTeX , the auxiliary files, the logical and visual design of an article, \LaTeX error messages. Finally, we present a long list of dos and don'ts to help you write good \LaTeX .

2.1 *Structure*

\LaTeX 's core is a programming language called \TeX , created by Donald E. Knuth, which provides low-level typesetting instructions. \TeX comes with a set of fonts called *Computer Modern* (CM). The CM fonts and the \TeX programming language form the foundation of a typical \TeX system. \TeX is extensible—new commands can be defined in terms of more basic ones. \LaTeX is one of the best known and most widely used extensions of \TeX .

The visual layout of a \LaTeX document is primarily determined by the *document*

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-55281-6_2



Figure 2.1: The structure of L^AT_EX.

class, such as `amsart` or `article` for articles and `amsbook` or `book` for books. Many journals, publishers, and schools have their own document classes for formatting articles, books, and theses.

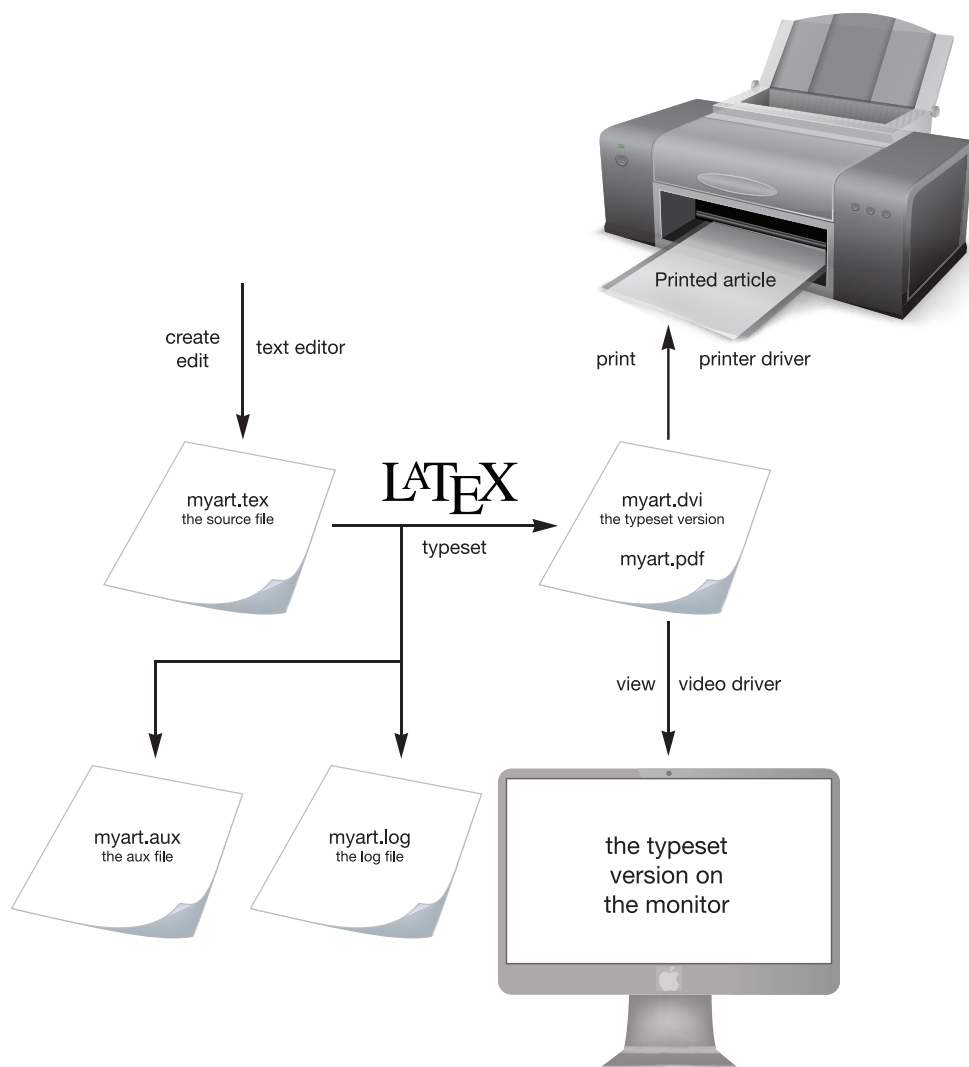
Extensions of L^AT_EX are called *packages*. They provide additional functionality by adding new commands and environments, or by changing the way previously defined commands and environments work. It is essential that you find the packages that make your work easier. LC3 discusses a large number of useful packages as of 2023.

The structure of L^AT_EX is illustrated in Figure 2.1. This figure suggests that in order to work with a L^AT_EX document, you first have to install T_EX and the CM fonts, then L^AT_EX, and finally specify the document class and the necessary packages. For math, the packages must include `amsmath`, `amsthm`, and so on. Of course, your L^AT_EX installation already has all of these.

2.2 Auxiliary files

Figure 2.2 illustrates the steps in the production of a typeset document.

You start by opening an existing L^AT_EX source file or creating a new one with a text editor. For this discussion, the source file is called `myart.tex`. Once the source file is ready, you typeset it. Depending on the document class options you choose and the packages the document loads, you end up with at least three additional files:

Figure 2.2: Using $L^A T_E X$.

1. `myart.pdf` The typeset article in PDF format.
2. `myart.aux` The auxiliary file, used by \LaTeX for internal bookkeeping, including cross-references and bibliographic citations.
3. `myart.log` The log file. \LaTeX records the typesetting session in the log file, including any warnings and messages that appear on your monitor in the log window.

Your computer uses a *video driver* to display the typeset article on your monitor and a *printer driver* to print the typeset article on a printer. The video and printer drivers are computer and \LaTeX implementation dependent.

It should be emphasized that of the four applications used (\TeX , video and printer drivers, editor), only one is the same for all computers and all implementations.

\LaTeX always uses the aux file from the last typesetting. Here is an example. Your article has Theorems 1 (with `\label{T:first}`) and 2 (with `\label{T:main}`). The aux file has the two lines:

```
\newlabel{T:first}{{1}{1}}
\newlabel{T:main}{{2}{1}}
```

`\newlabel{T:first}{{1}{1}}` means that the label `T:first` is assigned the value 1 and appears on page 1. `\newlabel{T:main}{{2}{1}}` means that the label `T:main` is assigned the value 2 and appears on page 1. So the reference

see Theorems `\ref{T:first}` and `\ref{T:main}`.

is typeset as

```
[ see Theorems 1 and 2.
```

Now add a new theorem between Theorems 1 and 2. Typeset the article. In the typeset article, the three theorems are properly numbered, but it still contains the same typeset line:

```
[ see Theorems 1 and 2.
```

The aux file has the lines:

```
\newlabel{T:first}{{1}{1}}
\newlabel{T:main}{{3}{1}}
```

So at the next typesetting, the reference is displayed as

```
[ see Theorems 1 and 3.
```

2.3 Logical and visual design

The typeset version of `art1.tex` looks impressive on p. 4. To produce such articles, you need to understand that there are two aspects of article design: *visual* and *logical*.

As an example, let us look at a theorem from `art1.tex` (see the typeset form of the theorem on page 4). You tell \LaTeX that you want to state a theorem by using a `theorem` environment:

```
\begin{theorem}\label{T:technical}  
Let  $L$  be a finite lattice.  
...  
\end{theorem}
```

The logical part of the design is choosing to define a theorem by placing material inside a `theorem` environment. For the visual design, \LaTeX makes hundreds of decisions. Could you have specified all of the spacing, font size changes, centering, numbering, and so on? Maybe, but would you *want* to? And would you want to repeat that process for every theorem in your document?

Even if you did, you would have spent a great deal of time and energy on the *visual design* of the theorem rather than on the *logical design* of your article. The idea behind \LaTeX is that you should concentrate on what you have to say and let \LaTeX take care of the visual design.

This approach allows you to easily alter the visual design by changing the document class (or its options; see Sections 11.5 and 19.1). Section 11.1 provides some examples. If you code the visual design into the article—hard coding it, as a programmer would say—such changes are much harder to accomplish, for you and for the journal publishing the article.

2.4 General error messages

Now that you are ready to type your first document, we give you some pointers on using \LaTeX .

You will probably make a number of mistakes in your first document. These mistakes fall into the following categories:

1. Typographical errors, which \LaTeX blindly typesets.
2. Errors in formulas or in the formatting of the text.
3. Errors in your instructions to \LaTeX , that is, in commands and environments.

Typographical errors can be corrected by viewing and spell checking the source file, finding the errors, and then editing the typeset file. Mistakes in the second and third categories may trigger errors during the typesetting process, such as the lines too wide of Section 1.4.

We now look at some examples of the third class of errors by deliberately introducing a number of mistakes into `art1.tex` and examining the messages.

Experiment 1. In `art1.tex`, go to line 18 (use the `Go to Line` command of your editor) and remove the closing brace so that it reads `\begin{abstract`

When you typeset `art1.tex`, \LaTeX reports a problem:

```
{abstract We present a technical result for congruences on\ETC.
./art1.tex:22:
Paragraph ended before \begin was complete.
<to be read again>
\par
```

1.22

Line 22 of the file is the line after `\maketitle`. The message informs you that the environment name was not completed.

Runaway argument? is a message that comes up often. It means that the argument of a command is either longer than expected or it contains material the argument cannot accept. Most often a closing brace solves the problem, as in this experiment.

Experiment 2. Now restore line 19, then go to line 20 and change `\end{abstract}` to `\end{abstrac}` and typeset again. \LaTeX informs you of another error:

```
./art1.tex:21: LaTeX Error: \begin{abstract}
on input line 18 ended by \end{abstrac}.
```

See the \LaTeX manual or \LaTeX Companion for explanation.
Type `H <return>` for immediate help.

...

```
1.20 \end{abstrac}
```

This is perfect. \LaTeX correctly analyzes the problem and tells you where to make the change. (Ignore `Type H <return> for immediate help.`, no help would be forthcoming.)

Experiment 3. Correct the error in line 21, and introduce a new error in line 60. This line reads

```
z_1 \equiv y+ w \pmod{\delta}.
```

Change `\delta` to `\deta`. Now, when you typeset the document, \LaTeX reports

```
./art1.tex:60: Undefined control sequence.
<argument> {\operator@font mod}\mkern 6mu\deta
```

```
1.60 z_1 \equiv y+ w \pmod{\deta}
```

This mistake is easy to identify: `\deta` is a misspelling of `\delta`.

Experiment 4. Correct the error, and in line 37, delete the closing brace of the `\label` command. This results in a message:

```
Runaway definition?
->E:cover\text {If  $x$  is covered by  $y, z$  in  $L$  and\ETC.
! File ended while scanning definition of \df@label.
<inserted text>
    }
<*> art1.tex
```

Undo the change to line 37.

$$z_1 \equiv y + w \pmod{\delta}.$$

Experiment 5. Correct the error, and add a blank line following line 60:

$$z_1 \equiv y \vee w \pmod{\delta}.$$

This change results in the message

```
./art1.tex:61: Missing $ inserted.
<inserted text>
    $
1.61
```

There can be no blank lines within a displayed math environment. \LaTeX catches the mistake, but the message itself is not user friendly.

Experiment 6. Correct the error, and add `$` before `\pmod` in line 60 (such an error often occurs when cutting and pasting a formula). You get the message:

```
./art1.tex:60: Display math should end with $$
<to be read again>
    \protect
1.60 z_1 \equiv y+ w $\pmod{\delta}
```

Maybe this could be more to the point?

This type of mistake happens a lot when you copy an inline formula into a displayed one.



Tip \LaTeX 's messages are not very useful with displayed formulas. Comment out lines to try to localize the problem. For many types of error messages, this is the best approach!



Tip Typeset often.

Typesetting my book *First Steps into L^AT_EX* with the closing brace of the first `\caption` command on line 480 of the source file missing, I get the error message

```
! Text line contains an invalid character.
1.1227 ...pletely irreducible^^?
```

where the reference is to line 1227, about 700 lines removed from the actual error. However, if the only thing I did before typesetting was to insert that figure with its incorrect caption command, at least, I would know where to look for errors. If you make a dozen changes and then typeset, you may not know where to start.

Error messages in L^AT_EX is a bit like getting error messages in Pascal in Apple Pages. Persevere, you do not have a choice.

2.5 Errors in math

Even in such a simple note as `formulanote.tex` there are opportunities for errors. To help familiarize yourself with some of the most commonly seen L^AT_EX errors in formulas, we introduce mistakes into `formulanote.tex`.

Experiment 1 In line 6 of `formulanote.tex`, delete the third `$` symbol; save the file in the work folder, and typeset it. L^AT_EX generates the following message:

```
! Missing $ inserted.
<inserted text>
      $
1.6 as $(u, v)$ and (u, \infty
                        )$. Such an interval
```

L^AT_EX reads `(u, \infty)` as text; but the `\infty` command instructs L^AT_EX to typeset a math symbol, which can only be done in a formula. So L^AT_EX offers to put a `$` in front of `\infty` while typesetting the source file—it does not put the `$` in the source file itself. L^AT_EX attempts a cure, but in this example it comes too late, because the formula *should* start just before `(u`.

Experiment 2 In line 16 of `formulanote.tex`, delete the second `}` symbol and save the file under the name `formulanotebad2.tex` in the work folder. This introduces an error: the closing brace of the first argument of `\frac` is missing. Now typeset the note. You get the message

```
Runaway argument?
{f(x) - f(a){x - a}, \] we assume that the function is defined
and co\ETC.
! File ended while scanning use of \frac .
<inserted text>
      \par
```

\LaTeX reports that a closing brace $\}$ is missing. To remedy this, you must look in the formula for an opening brace $\{$ that is not balanced, and insert the missing closing brace $\}$. Make the necessary change and typeset again to view the difference.

Experiment 3 In `formulanote.tex`, delete the two $\$$ signs in line 19, that is, replace $\$a\$$ by a . Typeset the file. It typesets with no errors. Here is the last line of the typeset file you get:

[we assume that the function is defined and continuous in a neighborhood of a .
[
instead of
[we assume that the function is defined and continuous in a neighborhood of a .
[

This is probably the error most often made by beginners. There is no message by \LaTeX and the typeset version looks good. Notice, however, the difference in the shape of the letter a in the two cases. You need sharp eyes to catch such an error.



Tip After an error is corrected, \LaTeX can refuse to typeset your document. If your document is `document.tex`, look in the same folder for the *auxiliary file* `document.aux` that was created by \LaTeX . Delete `document.aux` and **typeset twice**. See Section 2.2.

2.6 Your errors: Davey's Dos and Don'ts

Based on his many years of experience correcting \LaTeX articles for the journal *Algebra Universalis*, Brian Davey collected the \LaTeX mistakes most often made by authors. Here are some items from his list, divided into three categories.

Commands

1. Place ALL custom commands and environments in the preamble!
If you have trouble with custom commands, then you know where to find them.
2. Don't use `\def`; rather use `\newcommand` and `\renewcommand`.
`\def` is a \TeX command. It is like `\newcommand` (see Sections 1.9 and 15.1), but it can redefine an existing command. Redefining your own commands is bad enough, redefining a \TeX command can be a disaster.
3. Do not simply type the name of an operator into a formula. Declare the appropriate operator; see Sections 1.9 and 15.1.

For instance, do not type `$length I$`; it typesets as *lengthI*. It should be *length I*, typed as `$\length I$`. Of course, you have to add

```
\DeclareMathOperator{\length}{length}
```

to the preamble (see Section 1.8).

4. When you send a document to a coauthor or submit an article to a journal, remove all the custom commands not used.
This is a real time saver for your coauthor and editor.

Text

1. Do not produce a list with horizontal and vertical spacing commands. Use a list environment; see Sections 3.7 and 4.2.
2. Do not type numbers for citations and internal references. Use `\cite{...}` for citations and `\ref{...}` for references. For references to equations, use `\eqref`; see Sections 1.7.1 and 7.3.
3. Do not number proclamations (see Section 4.4). Use the standard `amsart` environments for theorems, and so on, and let L^AT_EX number them.
4. When writing a document for a journal requiring a document class file, **do not**
 - (a) change any of the size parameters: for instance, do not use options like `12pt` to change the font size or the `\setlength` command to change any parameter of the page size;
 - (b) do not insert vertical white space via `\bigskip`, `\smallskip`, `\vskip`, `\vspace`, etc., nor via your own custom commands. Do not adjust horizontal space without a very good reason.

So if you want to display some text:

┌

Please, display this text.

└

don't do this:

```
\medskip
\hspace*{6pt} Please, display this text.
\medskip
```

but rather

```
\begin{itemize}
\item[] Please, display this text.
\end{itemize}
```

```

or

\begin{quote}
Please, display this text.
\end{quote}

```

5. Do not leave a blank line before `\end{proof}` or before a text environment (see Section 4.1).
6. Do not use the `geometry` package.

Formulas

1. Do not leave a blank line before a displayed formula.
2. Don't use the symbol `|` in a set description, use the binary relation `\mid`; see Section 7.5.4.
For instance, `\{ x | x^2 < 2 \}` typesets as $\{x|x^2 < 2\}$. The correct form is $\{x \mid x^2 < 2\}$, typed as `\{x \mid x^2 < 2\}`.
3. Don't put punctuation marks inside an inline math environment.
For instance, `\sin x.` typed as `\sin x.$`; use `\sin x$`. This typesets as $\sin x$. Notice the smaller space between “ $\sin x$.” and “typed” and the wider space between “ $\sin x$$.” and “This”; see Sections 1.3 and 3.2.2.
4. Don't use two or more displayed formulas one after another. Use an appropriate environment such as `align`, `alignat`, `gather`, and so on (see Section 9.1.1).
5. Don't use `\left\{`, `\right\}`, `\left(`, `\right)`, and so on, by default (see page 16 and Section 7.5.1 for the commands `\left` and `\right`). Even when `\left` and `\right` do not change the size of the symbol, they add extra space after the closing delimiter.
6. Use `\colon` for functions. For instance, `f(x) \colon x \to x^2` typesets as $f(x): x \rightarrow x^2$. If you type `f(x) : x \to x^2`, you get $f(x) : x \rightarrow x^2$; the spacing is bad.
7. Use `\[` and `\]` (or `equation*`) to type a displayed math environment (see Section 1.7) rather than the old \TeX `$$` matched by `$$`. While display math produced via the latter does work properly most of the time, there are some \LaTeX commands that do not; for example, `\qedhere`.
8. Do not use the `center` environment to display formulas.
9. Use `\dots` first and let \LaTeX make the decision whether to use `\dots` or `\cdots`; see page 16 and Section 7.4.3. If \LaTeX gets it wrong, then use `\cdots` or `\ldots`.

10. If you can, avoid constructs (for instance, $\overset{\text{up}}{\rightarrow}$) in inline formulas that disrupt the regular line spacing. Although \LaTeX automatically leaves room for it, it does not look good, as a rule.

PART II

Text into L^AT_EX



Typing text

In Chapter 1, we briefly discussed how to type text in a document. Now we take up this topic more fully.

This chapter starts with a discussion of the keyboard in Section 3.1 and continues with the rules for spaces in Section 3.2. We cover in Section 3.3 a very important topic that must precede any in-depth discussion of \LaTeX : how to control \LaTeX with commands and environments.

A document may contain symbols that cannot be found on your keyboard. In Section 3.4, we show how to get these symbols in our typeset documents by using commands.

Some other characters are defined by \LaTeX as command characters. For example, the % character plays a special role in the source document. In Section 3.5.1, you will see how % is used to comment out lines and, in Section 3.5.2, we introduce the command for footnotes.

In Section 3.6, you learn about lines, paragraphs, and pages. The judicious use of horizontal and vertical spacing is an important part of document formatting, and also the topic of Section 3.7. In Section 3.8, you learn how to typeset text in a “box”, which

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-55281-6_3

behaves as if it were a single large character.

To help the discussion along, we shall use the terms *text mode* and *math mode* to distinguish between typesetting text and math.

3.1 The keyboard

Most of the keys on your computer's keyboard produce characters, while others are function or modifier keys.

3.1.1 Basic keys

The basic keys are grouped as follows:

Letters The 52 letter keys:

a b c ... z A B C ... Z

Digits The ten digits:

1 2 ... 9 0

Old-style digits are available with the `\oldstylenums` command. The next line shows the default digits followed by the old style digits:

1234567890 1234567890

typed as

1234567890 \quad \oldstylenums{1234567890}

Punctuation There are nine punctuation marks:

, ; . ? ! : ' ' -

The first six are the usual punctuation marks. The ' is the *left single quote*—also known as the *grave accent*—while ' doubles as the *right single quote* and *apostrophe* (see Section 3.4.1). The - key is the *dash* or *hyphen* (see Sections 3.4.2 and 3.4.9).

Parentheses There are four:

() []

(and) are *parentheses*; [and] are called (*square*) *brackets*.

Math symbols Seven math symbols correspond to keys. The math symbols are:

* + = - < > /

The last four characters have a role also in text mode:

- The minus sign `-` corresponds to the hyphen key, `-` (see Section 3.4.9).
- The math symbols `<` and `>` correspond to the keys `<` and `>`; use them only in math mode.

Note that there is also a version of colon `:` for math formulas (see Section 8.1 and Appendix A.2).

Space keys Pressing the space bar gives the *space character*. Pressing the tab key gives the *tab character*. When typesetting the source file, \LaTeX does not distinguish between these two. Pressing the *Return* key gives the *end-of-line character*. These keys produce *invisible characters* that are normally not displayed on your monitor by the text editor. Different computer systems have different end-of-line characters, which may cause some problems when transferring files from one system to another. A good text editor translates end-of-line characters automatically or on demand. Section 3.2.1 explains how \LaTeX handles the invisible characters.

When explaining some rules of \LaTeX , sometimes it is important to show if a space is required. In such cases, I use the symbol `_` to indicate a space, for instance, `\in_ut` and `_.`

The tilde `~` produces a *nonbreakable space* or *tie* (see Section 3.4.3).

3.1.2 Special keys

There are 13 special keys on the keyboard:

\$ % & ~ _ ^ \ { } @ " |

They are mostly used to give instructions to \LaTeX . Some are used in math mode (see Chapter 7), and some in \BibTeX (see Chapter 16). See Section 3.4.4 on how to print these characters in text. Only `@` requires no special command, type `@` to print `@`.

3.1.3 Prohibited keys

Keys other than those discussed in Sections 3.1.1 and 3.1.2 are prohibited! Some modern \LaTeX implementations will accept special characters that can be generated by various techniques, including the use of modifier keys such as Control, Alt, Escape, and typeset them correctly, but for the present at least, any documents which will be shared or submitted for publication should treat these as prohibited, as they may produce serious (and unnoticed) errors when typeset on another system. The `babel` package provides support for using some modifier keys.



Tip If there is a prohibited character in your document, you may receive a message such as

```
! Text line contains an invalid character.
1.222 completely irreducible^^?
                                ^^?
```

Delete and retype the offending word or line until the error goes away.

3.2 Words, sentences, and paragraphs

Text consists of words, sentences, and paragraphs. In text, *words* are separated by one or more spaces, which may include a single end-of-line character (see the rule, **Spacing in text**), or by parentheses and punctuation marks. A group of words terminated by a period, exclamation point, or question mark forms a *sentence* (not all periods terminate a sentence; see the discussion in Section 3.2.2). A group of sentences terminated by one or more blank lines constitutes a *paragraph*.

3.2.1 Spacing rules

Here are the most important L^AT_EX rules about spaces in text in the source file.



Rule: 1. Spacing in text.

Two or more consecutive spaces in text are the same as one.



Rule: 2. Spacing in text.

Two or more consecutive spaces in text are the same as one. A tab or end-of-line character is the same as a space.



Rule: 3. Spacing in text.

A blank line, that is, two end-of-line characters separated only by spaces and tabs, indicates the end of a paragraph. The `\par` command is equivalent.

**Rule: 4. Spacing in text.**

Spaces at the beginning of a line are ignored.

Rules 1 and 2 make cutting and pasting text less error-prone. In your source file, you do not have to worry about the line length or the number of spaces separating words or sentences, as long as there is at least one space or end-of-line character separating any two words. Thus

```
You  do not have to    worry
about the number of   spaces
separating words, as long as there
is  at least one space or end-of-line character
separating  any two words.
```

produces the same typeset text as

```
You do not have to worry about the number of spaces
separating words, as long as there is at least one space
or end-of-line character separating any two words.
```

However,

```
the number of   spaces separating words,
as long
```

and

```
the number of   spaces separating words
, as long
```

produce different results:

```
┌
the number of spaces separating words, as long
the number of spaces separating words , as long
└
```

Notice the space between “words” and the comma in the second line. That space was produced by the end-of-line character in accordance with Rule 2.

It is very important to maintain the readability of your source file. \LaTeX may not care about the number of spaces or line length, but you, your coauthor, and your editor might.

Rule 3 contradicts Rules 1 and 2, consider it an exception. Sometimes—especially when defining commands and environments (see Sections [15.1](#) and [15.2](#))—it is more convenient to indicate the end of a paragraph with `\par`.

3.2.2 Periods

\LaTeX places a certain size space between words—the *interword space*—and a somewhat larger space between sentences—the *intersentence space*. To know which space to use, \LaTeX must decide whether or not a period indicates the end of a sentence.



Rule: 1. Period.

To \LaTeX , a period after a capital letter, for instance, A. or caT., signifies an abbreviation or an initial. Generally, every other period signifies the end of a sentence.

This rule works most of the time. When it fails—for instance, twice with e.g.—you need to specify the type of space you want, using the following two rules.



Rule: 2. Period

If an abbreviation does not end with a capital letter, for instance, etc., and it is not the last word in the sentence, then follow the period by an interword space (\backslash_\square) or a tie (\sim), if appropriate (see Section 3.4.3).

Recall that \backslash_\square provides an interword space.

The result was first published, in a first approximation,
in the Combin.\ Journal. The result was first published,
in a first approximation, in the Combin. Journal.

prints as



The result was first published, in a first approximation, in the Combin. Journal.
The result was first published, in a first approximation, in the Combin. Journal.

Notice that Combin. in the first line is followed by a regular interword space. The intersentence space following Combin. in the second line is a little wider; it is an error.

A tie (or nonbreakable space)—see also Section 3.4.3—is more appropriate than \backslash_\square in phrases such as Prof. Smith, typed as Prof.~Smith, and pp. 271–292, typed as pp.~271--292.



Tip The `thebibliography` environment handles periods properly. You do not have to mark periods for abbreviations (in the form `._`) in the name of a journal, so

`Acta Math. Acad. Sci. Hungar.`

is correct.



Rule: 3. Period

If a capital letter is followed by a period and is at the end of a sentence, precede the period with `\@`.

For example,

`(1) follows from condition~H\@. We can proceed\\`

`(1) follows from condition~H. We can proceed`

prints:

┌

(1) follows from condition H. We can proceed

(1) follows from condition H. We can proceed

└

Notice that there is not enough space after H. in the second line.

Most typographers agree on the following rule (see, for instance, *The Elements of Typographic Style* by Robert Bringhurst [4], p. 30):



Rule: 4. Period

Add no space or a thin space (`\,`) within strings of initials and be consistent.

So W.H. Lampstone with no space or W.H. Lampstone with thin space is preferred over W. H. Lampstone. My personal choice is W. H. Lampstone with thin space.

To make all intersentence spaces equal to the interword space—as required in French typography—you can use the command

`\frenchspacing`

To switch back to using spaces of different sizes, give the command

`\nonfrenchspacing`

3.3 Commanding L^AT_EX

How do you command L^AT_EX to do something special for you, such as starting a new line, changing emphasis, or displaying the next theorem? You use *commands* and special pairs of commands called *environments*, both briefly introduced in Section 1.1.4.

Most, but not all, commands have *arguments*, which are usually fairly brief. Environments have *contents*, the text between the `\begin` and `\end` commands. The contents of an environment can be several paragraphs long.

3.3.1 Commands and environments

The `\emph{text}` command instructs L^AT_EX to emphasize its argument, *text*. The `\&` command has no argument; it instructs L^AT_EX to typeset `&` (see Section 3.4.4).

The `\flushright` environment instructs L^AT_EX to flush the content to the right, the text between the two commands

```
\begin{flushright}
\end{flushright}
```

The content of the `\document` environment is the body of the article (see Section 1.8) and the content of the `\abstract` environment is the abstract.



Rule: Environments

An environment starts with the command

```
\begin{name}
```

and ends with

```
\end{name}
```

Between these two commands is the *content* of the environment, affected by the definition of the environment.



Rule: Commands

A L^AT_EX command starts with a backslash, `\`, and is followed by the *command name*. The *name* of a command is either a *single non-alphabetic character* other than a tab or end-of-line character or a *string of letters*, that is, one or more letters.

So `#` and `'` are valid command names. The corresponding commands `\#` and `\'` are used in Sections 3.4.4 and 3.4.7, respectively. `\input` and `\date` are also valid command names. However, `\input3`, `\in#ut`, and `\in_ut` are not valid names because `3`, `#`, and `_`

□ should not occur in a multicharacter command name. Note that □ is a command name. The command \□ produces a blank.

L^AT_EX has a few commands, for instance, \$ (see Section 7.1) that do not follow this naming scheme, that is, they are not of the form \name. See also Section 15.1.9 for special commands with special termination rules.



Rule: Command termination

L^AT_EX finds the end of a command name as follows:

- If the first character of the name is not a letter, the name is the first character.
- If the first character of the name is a letter, the command name is terminated by the first nonletter.

If the command name is a string of letters, and is terminated by a space, then L^AT_EX discards all spaces following the command name.

While input3 is an invalid name, \input3 is not an incorrect command. It is the \input command followed by the character 3, which is either part of the text following the command or the argument of the command.

L^AT_EX also allows some command names to be modified with *. Such commands are referred to as **-ed commands*. Many commands have *-ed variants. \hspace* is an often-used *-ed command (see Section 3.7.1).



Rule: Command and environment names

Command and environment names are *case sensitive*. \ShowLabels is not the same as \showlabels.



Rule: Arguments

Arguments are enclosed in braces, { }, except for single character arguments. Optional arguments are enclosed in brackets, [].

Commands may have *arguments*, typed in braces immediately after the command. The argument(s) are used in processing the command.

Accents provide very simple examples. For instance, \'{o}—which produces ó—consists of the command \' and the argument o (see Section 3.4.7). In the command

```
\bibliography{article1}
```

the command is `\bibliography` and the argument is `article1` (see Section 16.2.2).

Sometimes, if the argument is a single character, the braces can be dropped: `\'o` also typesets as ó.

Some environments also have arguments. For example, the `alignat` environment (see Section 9.5.4) is delimited by the commands

```
\begin{alignat}{2}
```

and

```
\end{alignat}
```

The argument, 2, is the number of columns—it could be any number 1, 2, ... A command or environment may have more than one argument. The `\frac` command (see Section 7.4.1) has two; `$$\frac{1}{2}$$` typesets as $\frac{1}{2}$. The custom command `\cong` has three (see Section 15.1.2).

Some commands and environments have one or more *optional arguments*, that is, arguments that may or may not be present. The `\sqrt` command (see Section 7.4.5) has an optional argument for specifying roots other than the square root. To get $\sqrt[3]{25}$, type `\sqrt[3]{25}`. The `\documentclass` command has an argument, the name of a document class, and an optional argument, a list of options (see Section 10.2), for instance,

```
\documentclass[12pt,draft,leqno]{amsart}
```



Tip If you get an error when using a command, check that:

1. The command is spelled correctly, including the use of uppercase and lowercase letters.
2. You have specified all required arguments in braces.
3. Any optional argument is in brackets, not braces or parentheses.
4. The command is properly terminated.
5. The package providing the command is loaded with the `\usepackage` command.

Most errors in the use of commands are caused by breaking the termination rule. We can illustrate some of these errors with the `\today` command, which produces today's date. You have already seen this command in Section 1.3 (see also Section 3.4.8). The correct usage is

```
\today\ is the day
```

or

`\today{}` is the day

which both typeset in the following form

```
┌
January 19, 2024 is the day
└
```

In the first case, `\today` was terminated by `_`, the command that produces an inter-word space. In the second case, it was terminated by the *empty group* `{ }`.

If there is no space after the `\today` command, as in

`\todayis_the_day`

you get the message

```
! Undefined control sequence.
1.3 \todayis
      the day
```

\LaTeX thinks that `\todayis` is the command, and, of course, does not recognize it.

If you type one or more spaces after `\today`:

`\today_is_the_day`

\LaTeX interprets the two spaces as a single space by the first space rule (see page 50), and uses that one space to delimit `\today` from the text that follows it. So \LaTeX produces

```
┌
January 19, 2024is the day
└
```

Section 15.1.9 discusses how best to avoid such errors.



Rule: Optional argument

If a command—or environment—can have an optional argument and

- none is given, and
- the text following the command starts with `[`,

then type this as `{[]}`.

This may happen, for instance, with the command `\item` (see page 103). To get an example for an environment, see Section 9.6.1 for subsidiary math environments and page 233 for the alignment options. See what happens if no option is given but the math starts, say, with `[x]`.

3.3.2 Scope

A command issued inside a pair of braces `{ }` has no effect beyond the right brace, except for the rare *global* commands (see Section 3.3.3). You can have any number of pairs of braces:

```
{ ... { ... { ... } ... } ... }
```

The innermost pair containing a command is the *scope* of that command. The command has no effect outside its scope. We can illustrate this concept using the `\bfseries` command that switches the font to boldface:

```
{some text \bfseries bold text} no more bold
```

typesets as

```
┌
└ some text bold text no more bold
```

The commands `\begin{name}` and `\end{name}` bracketing an environment act also as a pair of braces. In particular, `$`, `\[`, and `\]` are also special braces.



Rule: Braces

1. Braces must be balanced: An opening brace has to be closed, and a closing brace must have a matching opening brace.
2. Pairs of braces cannot overlap.

Violating the first brace rule generates warnings and error messages. If there is one more opening brace than closing brace, the document typesets, but you get a warning:

```
(\end occurred inside a group at level 1)
```

For two or more unmatched opening braces, you are warned that `\end` occurred inside a group at level 2, and so on. There is a tendency to disregard such warnings since your article is already typeset and the unmatched opening brace may be difficult to find. However, such errors may have strange consequences. At one point in the writing of my second *L^AT_EX* book, there were two extra opening braces in Chapter 2. As a result, the title of Chapter 7 was placed on a page by itself! So it is best not to disregard such warnings.

If you have one unmatched closing brace, you get a message such as

```
! Too many }'s
```

If special braces, say, `\begin{name}` and `\end{name}`, do not balance, you get an error message:

```
! LaTeX Error: \begin{name} on input line 21
ended by \end{document}.
```

or

```
! LaTeX Error: \begin{document} ended by \end{name}.
```

To illustrate the second rule, here are two simple examples of overlapping braces.

Example 1

```
{\bfseries some text
\begin{lemma}
  more text} final text
\end{lemma}
```

Example 2

```
{some \bfseries text, then math:  $\sqrt{2}$  },  $\sqrt{3}$ }
```

In Example 1, the scope of `\bfseries` overlaps the `\begin{lemma}`, `\end{lemma}`. In Example 2, the scope of `\bfseries` overlaps the special braces `$` and `$`. Example 1 is easy to correct:

```
{\bfseries some text}
\begin{lemma}
  {\bfseries more text}
  final text
\end{lemma}
```

Example 2 may be corrected as follows:

```
{some \bfseries text, then math:}  $\sqrt{2}$ ,  $\sqrt{3}$ }
```

Actually, $\sqrt{2}$ does not even have a bold version (see Section 8.4.3).

If the braces do overlap and they are of the same kind, \LaTeX simply misunderstands the instructions. The closing brace of the first pair is regarded as the closing brace of the second pair, an error that may be difficult to detect. \LaTeX can help if special braces overlap. Typesetting Example 1 gives the message

```
! Extra }, or forgotten \endgroup.
1.7 more text }
      final text
```

3.3.3 Types of commands

It may be useful at this point to note that commands can be of various types.

Some commands have arguments, and some do not. Some commands effect change only in their arguments, while some commands declare a change.

For instance, `\textbf{This is bold}` typesets the phrase **This is bold** in bold type: **This is bold** and has no effect on the text following the argument of the command. On the other hand, the command `\bfseries` declares that the text that follows should be bold. This command has no argument. I call a command that declares change a *command declaration*. So `\bfseries` is a command declaration, while `\textbf` is not. As a rule, command declarations are commands without arguments.

Commands with arguments are called *long* if their argument(s) can contain a blank line or a `\par` command; otherwise they are *short*. For example, `\textbf` is a short command. So are all the top matter commands discussed in Section 11.2. The `\parbox` command, discussed in Section 3.8.4, is long.

Finally, as discussed in Section 3.3.2, the effect of a command remains within its scope. This is true only of *local* commands. There are also some *global* commands, such as the `\setcounter` command described in Section 15.5.1.

Fragile commands

As a rule, \LaTeX reads a paragraph of the source file, typesets it, and then goes on to the next paragraph. Some information from the source file, however, is separately stored for later use. Examples include the title of an article, which is reused as a running head (Section 11.2.1); titles of parts, sections, subsections, and other sectioning commands, which are used in the tables of contents (Sections 19.2 and 10.4.1); footnotes (Section 3.5.2); table and figure captions (Section 10.4.3), which are used in lists of tables and figures (Section 10.4.3); and index entries (Chapter 18).

These are *movable arguments*, and certain commands embedded in them must be protected from damage while being moved. \LaTeX commands that need such protection are called *fragile*. The inline math delimiter commands `\(` and `\)` are fragile, while `$` is not.

In a movable argument, fragile commands must be protected with a `\protect` command. Thus

The function `\(f(x^{\{2\}}) \)`

is not an appropriate section title, but

The function `\protect \(f(x^{\{2\}}) \protect \)`

is. So is

The function `$f(x^{\{2\}})$`

To be on the safe side, you should protect every command that might cause problems in a movable argument. Section 19.2 shows an example of what happens if a fragile command is not protected. Alternatively, use commands declared with

```
\DeclareRobustCommand
```

This command works the same way as `\newcommand` but the command defined is *robust*, that is, not fragile.

3.4 Symbols not on the keyboard

A typeset document may contain symbols that cannot be typed. Some of these symbols may even be available on the keyboard but you are prohibited from using them (see Section 3.1.3). In this section, we discuss the commands that typeset some of these symbols in text.

3.4.1 Quotation marks

To produce single and double quotes, as in

```
┌
└ ‘subdirectly irreducible’ and “subdirectly irreducible”
```

type

```
‘subdirectly irreducible’ and ‘‘subdirectly irreducible’’
```

Here, ‘ is the left single quote and ’ is the right single quote. Note that the double quote is obtained by pressing the single quote key twice, and *not* by using the double quote key. If you need single and double quotes together, as in “She replied, ‘No.’”, separate them with \, (which provides a thin horizontal space, see Section 3.7.1):

```
‘‘She replied, ‘No.’\,’’,
```

which typesets as “She replied, ‘No.’”,

3.4.2 Dashes

Dashes come in three lengths. The shortest dash, called a *hyphen*, is used to connect words:

```
┌
└ Mean-Value Theorem
```

This phrase is typed with a single dash:

```
Mean-Value Theorem
```

A medium-sized dash, called an *en dash*, is typed as `--` and is used:

- For number ranges; for instance, the phrase see pages 23–45, is typed as

see pages~23--45

Note: `~` is a nonbreakable space or tie (see Section 3.4.3).

- In place of a hyphen in a compound adjective when one of the elements of the adjective is an open compound (such as New York) or hyphenated (such as non-English). For instance, the phrase Schmidt–Freid adjoint, is typed as

Schmidt--Freid adjoint

A long dash—called an *em dash*—is used to mark a change in thought or to add emphasis to a parenthetical clause, as in this sentence. The two em dashes in the last sentence are typed as follows:

A long dash---called an `\emph{em dash}`---is used

In math mode, a single dash is typeset as the minus sign $-$ (a binary operation) with some spacing on both sides, as in $15 - 3$ or the “negative” as in -3 ; this is discussed in Sections 3.1.1 and 7.4.1.

Note that, as a rule, there is no space before or after an en dash or em dash.

3.4.3 Ties or nonbreakable spaces

A *tie* or *nonbreakable space* or *blue space* is an interword space that cannot be broken across lines. For instance, when referencing P. Neukomm in an article, you do not want the initial P. at the end of a line and the surname Neukomm at the beginning of the next line. To prevent such an occurrence, you should type `P.~Neukomm`.

If your keyboard does not have `~`, use the `\nobreakspace` command instead, and type `P.\nobreakspace Neukomm`. The following examples show some typical uses:

Theorem~\ref{T:main} in Section~\ref{S:intro}

Donald~E. Knuth

assume that $f(x)$ is (a)~continuous, (b)~bounded

the lattice~ \mathbb{L}

Sections~\ref{S:modular} and~\ref{S:distributive}

In~ \mathbb{L} , we find

Name	Type	Typeset
Ampersand	<code>\&</code>	<code>&</code>
Caret	<code>\^{}{}</code>	<code>^</code>
Dollar Sign	<code>\\$</code>	<code>\$</code>
Left Brace	<code>\{</code>	<code>{</code>
Right Brace	<code>\}</code>	<code>}</code>
Underscore (or Lowline)	<code>_</code>	<code>-</code>
Octothorp	<code>\#</code>	<code>#</code>
Percent	<code>\%</code>	<code>%</code>
Tilde	<code>\~{}{}</code>	<code>~</code>

Table 3.1: Nine special characters.

Of course, if you add too many ties, as in

Peter~G.~Neukomm% Incorrect!

L^AT_EX may send you a line too wide message (see Sections 1.4 and 3.6.1).

The tie (~) absorbs spaces, so typing P.~_Neukomm works just as well. This feature is convenient when you add a tie during editing.

3.4.4 Special characters

The characters corresponding to nine of the 13 special keys (see Sections 1.2 and 3.1.2) are produced by typing a backslash (\) and then the key, as shown in Table 3.1.

If for some reason you want to typeset a backslash in your document, type the command `\textbackslash`, which typesets as `\`. You might think that you could get a typewriter style backslash by utilizing the `\texttt` command introduced in Section 1.3

`\texttt{\textbackslash}`

but this is not the case, `\textbackslash` and `\texttt{\textbackslash}` produce the same symbol, `\`, which is different from the typewriter style backslash: `\`. Look at them side-by-side: `\` `\`. For a typewriter style backslash you can use the `\bsl` command introduced in Section 15.1.1 or the `\texttt{\symbol{92}}` command introduced later in this section.

The `|` key is seldom used in text. If you need to typeset the math symbol `|` in text, type `\textbar`.

Note that in text, `*` typesets as `*`, whereas in a formula it typesets centered as `*`. To typeset a centered star in text, use the command `\textasteriskcentered`. And `@` typesets as `@`.

Finally, the `"` key should never be used in text. See Section 3.4.1 for the proper way to typeset double quotes (see also Section 1.3). Nevertheless, sometimes `"` may

be used to typeset ", as in the computer code segment `print("Hello!")`. In \LaTeX and *MakeIndex*, " has special meanings (see Chapters 16 and 18).



Tip Be careful when typing `\{` and `\}` to typeset the braces `{ }`. Typing a brace without its backslash results in unbalanced braces, in violation of the first brace rule in Section 3.3.2.

We illustrated in Section 3.3.2 some consequences of unbalanced braces. You may avoid some of these problems by introducing custom commands, as introduced in Section 15.3.

You can also produce special characters with the `\symbol` command:

```
\symbol{94} typesets as ^
\symbols{126} typesets as ~
\texttt{\symbol{92}} typesets as \
```

The argument of the `\symbol` command is a number matching the slot of the symbol in the layout (encoding) of the font. The layout for the Computer Modern typewriter style font is shown in Table 3.2.

Alternatively, instead of `\texttt{\symbol{92}}`, we can use

```
\texttt{\char'\}
```

Any character *x* in the font can be accessed by typing the character itself as `'\x`. This way you don't have to look up the position of the symbol.

You can obtain similar tables for any font in your \LaTeX implementation by using the `fonttbl.tex` file in your `samples` folder. The table format in this file is used in Section 4.6 as an example of the `tabular` environment.

3.4.5 Ellipses

The text ellipsis, ..., is produced using the `\dots` command. Typing three periods produces ... (notice that the spacing is wrong). `\dots` is one of several commands that can be used to create ellipses in formulas (see Section 7.4.3).

3.4.6 Ligatures

Certain groups of characters, when typeset, are joined together—such compound characters are called *ligatures*. There are five ligatures that \LaTeX typesets automatically (if you use the Computer Modern fonts): `ff`, `fi`, `fl`, `ffi`, and `ffl`.

If you want to prevent \LaTeX from forming a ligature, separate the characters with the command `\textcompwordmark`. Compare `iff` with `iff`, typed as `iff` and

	0	1	2	3	4	5	6	7	8	9
x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Τ	Φ	Ψ
1x	Ω	↑	↓	'	ı	ı	ı	ı	ı	ı
2x	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘
3x	Ɛ	∅	□	!	"	#	\$	%	&	'
4x	()	*	+	,	-	.	/	0	1
5x	2	3	4	5	6	7	8	9	:	;
6x	<	=	>	?	@	A	B	C	D	E
7x	F	G	H	I	J	K	L	M	N	O
8x	P	Q	R	S	T	U	V	W	X	Y
9x	Z	[\]	ˆ	˘	˘	a	b	c
10x	d	e	f	g	h	i	j	k	l	m
11x	n	o	p	q	r	s	t	u	v	w
12x	x	y	z	{		}	~	ˆ	˘	˘

Table 3.2: Font table for the Computer Modern typewriter style font.

```
if\textcompwordmark f
```

Enclosing the second character in braces (`{}`) is a crude method of preventing the ligature, as used in Formula 4 of the *Formula Gallery* (see Section 7.10). This method, in some instances, may interfere with \LaTeX 's hyphenation algorithm. See Section 15.1.1 for the custom command `\If`.

3.4.7 Accents and symbols in text

\LaTeX provides 15 European accents. Type the command for the accent (`\` and a character), followed by the letter (in braces) on which you want the accent placed (see Table 3.3). For example, to get Grätzer György, type

```
Gr\ "atzer Gy\ "orgy
```

and to get Ö type `\ "0`. For Grätzer György you should not type

```
Gr\ "{a}tzer Gy\ "{o}rgy
```

because it breaks the kerning.

To place an accent on top of an `i` or a `j`, you must use the *dotless* version of `i` and `j`. These are obtained by the commands `\i` and `\j`: `\' {\i}` typesets as `í` and `\v {\j}` typesets as `ĵ`. Tables 3.4 and 3.5 list some additional text symbols and European characters available in \LaTeX when typing text. Using localized versions of \LaTeX , you get more accented and combined characters such as the Catalan geminated ell.

Note that the `\textcircled` command (in Table 3.5) takes an argument. It seems to work best with a single lowercase character, like `@` or `@`. Capitals such as `A` are not very satisfactory. Section 3.8.6 explains how to create the symbol `A`.

3.4.8 Logos and dates

`\TeX` produces \TeX , `\LaTeX` produces \LaTeX , and `\LaTeX2e` produces $\LaTeX2e$ (the original name of the current version of \LaTeX). The `\AmS` command produces the logo $\mathcal{A}\mathcal{M}\mathcal{S}$.

Remember to type `\TeX\` or `\TeX{}` if you need a space after \TeX (similarly for the others). A better way to handle this problem is discussed in Section 15.1.1.

\LaTeX also stores some useful numbers:

- `\time` is the time of day in minutes since midnight,
- `\day` is the day of the month,
- `\month` is the month of the year,
- `\year` is the current year.

You can include these numbers in your document by using the `\the` command:

Year: `\the\year`; month: `\the\month`; day: `\the\day`

produces a result such as

```

┌
Year: 2023; month: 7; day: 11
└

```

Of more interest is the `\today` command, which produces today's date in the form: July 11, 2023. It is often used as the argument of the `\date` command (see Section 11.2.1).

Remember the termination rule (Rule 3 in Section 3.3.1).

today's date in the form: `\today` (you may want

Name	Type	Typeset	Name	Type	Typeset
acute	<code>\' {o}</code>	ó	macron	<code>\={o}</code>	ō
breve	<code>\u {o}</code>	ö	overdot	<code>\. {g}</code>	ḡ
caron/haček	<code>\v {o}</code>	ǎ	ring	<code>\r {u}</code>	ũ
cedilla	<code>\c {c}</code>	ç	tie	<code>\t {oo}</code>	ôo
circumflex	<code>\^ {o}</code>	ô	tilde	<code>\~ {n}</code>	ñ
dieresis/umlaut	<code>\" {u}</code>	ü	underdot	<code>\d {m}</code>	ṁ
double acute	<code>\H {o}</code>	Ő	underbar	<code>\b {o}</code>	ō
grave	<code>\' {o}</code>	ò			
dotless i	<code>\i</code>	ı	dotless j	<code>\j</code>	ȷ
	<code>\' {\i}</code>	í		<code>\v {\j}</code>	ȶ

Table 3.3: European accents.

Name	Type	Typeset	Type	Typeset
a-ring	<code>\aa</code>	å	<code>\AA</code>	Å
aesc	<code>\ae</code>	æ	<code>\AE</code>	Æ
ethel	<code>\oe</code>	œ	<code>\OE</code>	Œ
eszett	<code>\ss</code>	ß	<code>\SS</code>	SS
inverted question mark	<code>?‘</code>	¿		
inverted exclamation mark	<code>!‘</code>	¡		
slashed L	<code>\l</code>	ł	<code>\L</code>	Ł
slashed O	<code>\o</code>	ø	<code>\O</code>	Ø

Table 3.4: European characters.

Name	Type	Typeset
ampersand	<code>\&</code>	&
asterisk bullet	<code>\textasteriskcentered</code>	*
backslash	<code>\textbackslash</code>	\
bar (caesura)	<code>\textbar</code>	
brace left	<code>\{</code>	{
brace right	<code>\}</code>	}
bullet	<code>\textbullet</code>	•
circled a	<code>\textcircled{a}</code>	Ⓐ
circumflex	<code>\textasciicircum</code>	^
copyright	<code>\copyright</code>	©
dagger	<code>\dag</code>	†
double dagger (diesis)	<code>\ddag</code>	‡
dollar	<code>\\$</code>	\$
double quotation left	<code>\textquotedblleft</code> or ‘‘	“
double quotation right	<code>\textquotedblright</code> or ’’	”
em dash	<code>\textemdash</code> or ---	—
en dash	<code>\textendash</code> or --	–
exclamation down	<code>\textexclamdown</code> or !‘	¡
greater than	<code>\textgreater</code>	>
less than	<code>\textless</code>	<
lowline	<code>_</code>	-
midpoint	<code>\textperiodcentered</code>	·
octothorp	<code>\#</code>	#
percent	<code>\%</code>	%
pilcrow (paragraph)	<code>\P</code>	¶
question down	<code>\textquestiondown</code> or ?‘	¿
registered trademark	<code>\textregistered</code>	®
section	<code>\S</code>	§

Table 3.5: Extra text symbols.

produces

```
┌
today's date in the form: July 11, 2023(you may want
└
```

To get the desired effect, type `_` or `{}` after the `\today` command:

```
today's date in the form: \today\_ (you may want
```

3.4.9 Hyphenation

\LaTeX reads the source file one line at a time until it reaches the end of the current paragraph and then tries to balance the lines. To achieve this goal, \LaTeX hyphenates long words using a built-in hyphenation algorithm, a database stored in the `hyphen.tex` file, and a long `\hyphenation` list in the AMS document classes. If you use a document class not containing such a list, copy the hyphenation list from `amsart.cls` to your document.



Rule: Optional hyphen

If you find that \LaTeX cannot properly hyphenate a word, put *optional hyphens* in the word. An optional hyphen is typed as `\-`, and *allows* \LaTeX to hyphenate the word where the optional hyphen is placed—and only at such points—if the need arises.

Examples: `data_base`, `an_ti_thet_ic`, `set_up`

Note that:

- Optional hyphens prevent hyphenation at any other point in the word.
- Placing an optional hyphen in a particular occurrence of a word does not affect the hyphenation of any other occurrences of that word.



Rule: Hyphenation specifications

List the words that often need help in a command:

```
\hyphenation{set-up as-so-ciate}
```

All occurrences of the listed words following this command in your document are hyphenated as specified. Preferably, place it in the Preamble.

Note that in the `\hyphenation` command the hyphens are designated by `-` and not by `\-`, and that the words are separated by spaces not by commas.

You must use optional hyphens for words with accented characters, as in

`Gr\{"a}t\~zer`

Such words cannot be included in a `\hyphenation` list (unless you use the T1 font encoding—see Chapter 6).



Rule: Preventing hyphenation

To *prevent* hyphenation of a word, put it in the argument of an `\mbox` command (see Section 3.8) or place it unhyphenated in a `\hyphenation` command.

For example, type

`\mbox{database}`

if you do not want this instance of `database` hyphenated, or type

`\hyphenation{database}`

if you do not want \LaTeX to hyphenate any occurrence of the word after this command in your document. Of course, typing `data\~base` overrides the general prohibition for this one instance.

You can have any number of `\hyphenation` commands in your document, preferably, in the Preamble.

If the `amsmath` package was loaded (for instance, if you use the `amsart` document class), then you can use `\text` instead of `\mbox`:

`\text{database}`



Tip \LaTeX does not break a hyphenated word except at the hyphen, nor does it break a word followed by or preceding an em dash or en dash (see Section 3.4.2). \LaTeX often needs help with such words.

Sometimes a hyphen in a phrase should not be broken. For instance, the phrase `m-complete lattice` should not be broken after `m`; so type it as

`\mbox{\$ \mathfrak{m} \$-com} \~plete lattice`

(see Section 8.4.2 for `\mathfrak`).

Use the `\nobreakdash` command (placed before the hyphen)

`\nobreakdash- \nobreakdash-- \nobreakdash---`

to prevent such breaks.

For example,

```
pages~24\nobreakdash--47
```

Since \LaTeX does not hyphenate a hyphenated word except at the hyphen,

```
\nobreakdash-
```

prevents the hyphenation of the whole word as though it were enclosed in an `\mbox` command. The form

```
\nobreakdash-\hspace{0pt}
```

allows the normal hyphenation of the word that follows the hyphen. For example,

```
 $\frac{m}{n}$ \nobreakdash-\hspace{0pt}complete lattice
```

allows the word `complete` to be hyphenated.

This coding of the phrase `m-complete lattice` is a natural candidate for a custom command (see Section 15.1.1).



Tip If you want to know how \LaTeX would hyphenate a list of words, place it in the argument of a `\showhyphens` command.

For instance,

```
\showhyphens{summation reducible latticoid}
```

The result,

```
sum-ma-tion re-ducible lat-ti-coid
```

is shown in the log file.



Tip Some text editors wrap lines in a source file by breaking them at a hyphen, introducing errors in your typeset document.

For instance,

```
It follows from Theorem~\ref{T:M} that complete-
simple lattices are very large.
```

is typeset by \LaTeX as follows:



It follows from Theorem 2 that complete- simple lattices are very large.

As you can see, there is a space between the hyphen and the word simple. The text editor inserted an end-of-line character after the hyphen (by the second space rule; see Section 3.2.1). This end-of-line character was interpreted by \LaTeX as a space. To correct the error, make sure that there is no such line break, or comment out (see Section 3.5.1) the end-of-line character:

```
It follows from Theorem~\ref{T:M} that complete-%
simple lattices are very large.
```

Better yet, rearrange the two lines:

```
It follows from Theorem~\ref{T:M} that
complete-simple lattices are very large.
```

Of course, \LaTeX does not know everything about the complicated hyphenation rules of the English language. Consult *The Chicago Manual of Style*, 17th edition [6], H. W. Fowler, *A Dictionary of Modern English Usage* [10], and Lyn Dupré's *BUGS in Writing: A Guide to Debugging Your Prose*, 2nd edition [9] for additional guidance.

Barbara Beeton maintains a list of words incorrectly hyphenated by \TeX . Your \TeX distribution has this information.

3.5 Comments and footnotes

Various parts of your source file do not get typeset like most of the rest. The two primary examples are comments that do not get typeset at all and footnotes that get typeset at the bottom of the page.

3.5.1 Comments

The % symbol tells \LaTeX to ignore the rest of the line. A common use might be a comment to yourself in the source file:

```
therefore, a reference to Theorem~1 % check this!
```

The % symbol has many uses. For instance, a document class command (see Section 11.5),

```
\documentclass[twocolumn,twoside,legalpaper]{amsart}
```

may be typed with explanations, as

```
\documentclass[%
twocolumn,% option for two-column pages
twoside,% format for two-sided printing
legalpaper% print on legal-size paper
]{amsart}
```

so you can easily comment out some at a later time, as in

```
\documentclass[%
twocolumn,% option for two-column pages
%twoside,% format for two-sided printing
%legalpaper% print on legal-size paper
]{amsart}
```

Notice that the first line is terminated with a % to comment out the end-of-line character.



Tip Some command arguments do not allow any spaces in some T_EX distributions. If you want to break a line within an argument list, you can terminate the line with a %, as in the previous example.

See also the example at the end of Section 3.4.9.

It is often useful to start a document with a comment line giving the file name and identifying the earliest version of L^AT_EX that must be used to typeset it.

```
%This is article.tex
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
```

The second line specifies the December 1, 1994 (or later) release of L^AT_EX. You may need to use this declaration if your document incorporates features that were not available in earlier releases. (L^AT_EX is changing very little these days; maybe, \NeedsTeXFormat is becoming obsolete.)



Tip Use of % include marking parts of the article for your own reference. For instance, you may include comments to explain command definitions (as in Section 15.3). If something goes wrong inside a multiline math display (see Chapter 9), L^AT_EX does not tell you precisely where the error is. You can try commenting out all but one of the lines, until each line works separately.

Note that % does not comment out lines in a B_IB_TE_X database document (see Section 16.2.5).



Tip Symbolic referencing With every \label command I add the commented out form of the symbolic reference, see Section 1.7.2.

So if I start a new theorem, I type \the and my text expander inserts the following in the article:

```
\begin{theorem}\label{T:xx}
%Theorem~\ref{T:xx}

\end{theorem}
```

And do remember:



Rule: The 25% rule

If you want a % sign in text, make sure you type it as \%. Otherwise, % comments out the rest of the line. L^AT_EX does not produce a warning.

Using % to comment out large blocks of text can be tedious even with block comment. The `verbatim` package includes the `comment` environment:

```
\begin{comment}
...the commented out text...
\end{comment}
```

The `\end{verbatim}` command must be on a new line.



Rule: comment environments

1. `\end{comment}` must be on a line by itself.
 2. There can be no comment within a comment.
-

In other words,

```
\begin{comment}
  commented out text...
  \begin{comment}
    some more commented out text...
  \end{comment}
  and some more commented out text...
\end{comment}
```

is not allowed. L^AT_EX may give one of several messages, depending on the circumstances. For instance,

```
! LaTeX Error: \begin{document} ended by \end{comment}.
```

```
1.175 \end{comment}
```

**Tip Locating errors**

The `comment` environment can be very useful in locating errors.

Suppose you have unbalanced braces in your source file (see Section 3.3.2). Working with a *copy* of your source file, comment out the first half at a safe point (not within an environment!) and typeset. If you still get the same message, the error is in the second half. If there is no error message, the error is in the first half. Comment out the half that has no error.

Now comment out half of the remaining text and typeset again. Check to see whether the error appears in the first half of the remaining text or the second. Continue applying this method until you narrow down the error to a paragraph that you can inspect visually.

Since the `comment` environment requires the `verbatim` package, you must include the line

```
\usepackage{verbatim}
```

in the preamble of the source file (see Section 1.8).

3.5.2 Footnotes

The text of a footnote is typed as the argument of a `\footnote` command. To illustrate the use of footnotes, I have placed one here.¹ This footnote is typed as

```
\footnote{Footnotes are easy to place.}
```

If you want to use symbols to designate the footnotes, instead of numbers, type the command

```
\renewcommand{\thefootnote}
{\ensuremath{\fnsymbol{footnote}}}
```

before the first footnote; this provides up to nine symbols. In Section 15.1.1, we discuss the `\ensuremath` command. Section 3.5 of LC3 describes how to further customize footnotes.

In addition, there are title-page footnotes, such as the `\thanks` and `\date` commands in the top matter. See page 4 for a typeset example of `\date`. See also Section 11.2 and the typeset title page footnotes on page 299.

You can add a footnote marked by `*` to the title of an article. For instance, type the title

```
\title[Complete congruence lattices]%
{Complete congruence lattices$^{*}$}
```

¹Footnotes are easy to place.

and add the lines

```
{\renewcommand{\thefootnote}{\fnsymbol{footnote}}
\setcounter{footnote}{1}
\footnotetext{Lecture delivered at the \AmS\
              annual meeting in Brandon.}
\setcounter{footnote}{0}
}
```

The footnote will appear as the first footnote on page 1 marked by *. All the other footnotes are unmarked.

3.6 Lines, paragraphs, and pages

When typesetting a document, \LaTeX breaks the text into lines, paragraphs, and pages. Sometimes you may not like how \LaTeX has chosen to lay out your text. There are ways how to influence how \LaTeX does its work and these are discussed in this section.

3.6.1 Lines

We have not formally introduce the `\hbox` command: it is the \TeX version of the `\mbox` command; see Section 3.8. Here is a typical example.

\LaTeX typesets a document one paragraph at a time. It tries to split the paragraph into lines of equal width; for a preliminary discussion; see Section 1.4. If it fails to do so successfully and a line is too wide, you get an overfull `\hbox` message.

```
Overfull \hbox (15.38948pt too wide) in paragraph
      at lines 11--16
[]\OT1/cmr/m/n/10 In sev-eral sec-tions of the course
      on ma-trix
the-ory, the strange term ‘‘hamiltonian-
```

The log file records these messages. To place a visual warning in the typeset version of your document as well, use the draft document class option

```
\documentclass[draft]{amsart}
```

So the following line:

```
Lines that are too wide are marked
with a \emph{slug}
```

typesets as

Lines that are too wide are marked with a *slug* on the margin; it looks like this: ■ (it is a vertical bar of width `\overfullrule`).

Do not worry about such messages while writing the document. If you are preparing the final version and receive a message for an overfull `\hbox`, the first line of defense is to see whether optional hyphens would help (see Section 3.4.9). Read the warning message carefully to see which words \LaTeX cannot hyphenate properly. If adding optional hyphens does not help, a simple rephrasing of the problem sentence often does the trick.

Recall that there are 72.27 points in an inch (see Section 1.4). So if the message indicates a 1.55812 pt overflow, for instance, you can safely ignore it.



Tip If you do not want the 1.55812pt overflow reported whenever the document is typeset, you can enclose the offending paragraph (including the blank line indicating the end of the paragraph) between the lines

```
{\setlength{\hfuzz}{2pt}
```

```
and
```

Choose an argument that is slightly more than the reported error (maybe 2pt). This does not affect the typeset output, but the warning message and the slug, if you are using the draft option, are suppressed.

Alternatively, enclose the offending paragraph including the blank line indicating the end of the paragraph in a `setlength` environment:

```
\begin{setlength}{\hfuzz}{2pt}
\end{setlength}
```

Breaking lines

There are two forms of the line breaking command:

- The `\` and `\newline` commands break the line at the point of insertion but do not stretch it.
- The `\linebreak` command breaks the line at the point of insertion and stretches the line to make it of the normal width.

The text following any of these commands starts at the beginning of the next line, without indentation. The `\` command is often used, but `\linebreak` is rarely seen. (See Section 19.5 for an application of the `\linebreak` command.) I illustrate the effect of these commands:

There are two forms of the line breaking command:

There are two forms\\ of the line breaking command:

There are two forms \newline of the line breaking command:

There are two forms \linebreak of the line breaking command:

typeset as

┌

There are two forms of the line breaking command:

There are two forms

of the line breaking command:

There are two forms

of the line breaking command:

There are two forms

of the line breaking command:

└

If you force a line break in the middle of a paragraph with the `\linebreak` command and \LaTeX thinks that there is too little text left on the line to stretch it to full width, you get a message such as

Underfull \hbox (badness 4328) in paragraph

at lines 8--12

The `\\` command has two important variants:

- `\\[length]`, where *length* is the interline space you wish to specify after the line break, for instance, 12pt, .5in, or 1.2cm. Note how the units are abbreviated.
- `*`, which prohibits a page break following the line break.

The `*[length]` form combines the two variants. We illustrate the `\\[length]` command:

It is also semimodular.\\[15pt] In particular,

which is typeset as

┌

It is also semimodular.

└

In particular,

Since `\\` can be modified by `*` or by `[]`, \LaTeX may get confused if the line after a `\\` command starts with a `*` or `[`. In such cases, type `*` as `{*}` or `[` as `{[}`. For instance, to get

```

┌
There are two sources of problems:
[a] The next line starts with [.
└

type

There are two sources of problems:\\
{[a] The next line starts with \texttt{[.

If you fail to type {[}, you get the message

! Missing number, treated as zero.
<to be read again>

                                a

1.16 [a]
      The next line starts with \texttt{[.

```



**Rule: **

Without optional arguments, the \\ command and the \newline command are the same *in text*, but not within environments or command arguments.

You can qualify the \linebreak command with an optional argument: 0 to 4. The higher the argument, the more it forces the occurrence of a line break. The command \linebreak[4] is the same as \linebreak, while \linebreak[0] allows the line break but does not force it.

The \nolinebreak command plays the opposite role. \nolinebreak[0] = \linebreak[0], and \nolinebreak[4] = \nolinebreak. \nolinebreak is seldom used since the tie (~) and the \text command (see Section 3.4.3) accomplish the same goal most of the time.

Double spacing

It is convenient to proofread documents double spaced. Some journals even require submissions to be double spaced.

To typeset a document double spaced, include the command

```
\renewcommand{\baselinestretch}{1.5}
```

in its preamble.

Alternatively, use George D. Greenwade's `setspace`. Load this package with a

```
\usepackage{setspace}
```

command in the preamble of the document and specify

`\doublespacing`

in the preamble. This changes not just the line spacing but a number of other parameters to make your article look good.

There is also a

`\onehalfspacing`

command.

3.6.2 Paragraphs

Paragraphs are separated by blank lines or by the `\par` command. \LaTeX error messages always show paragraph breaks as `\par`. The `\par` form is also very useful in custom commands and environments (see Sections 15.1 and 15.2).

In some document classes, the first line of a paragraph is automatically indented. Indentation can be prevented with the `\noindent` command and can be forced with the `\indent` command.

Sometimes—for instance, in a schedule, glossary, or index—you may want a *hanging indent*, where the first line of a paragraph is not indented, and all the others are indented by a specified amount.

Hanging indents are created by specifying the amount of indentation specified by `\hangindent` and set with the `\setlength` command:

```
\setlength{\hangindent}{30pt}
```

```
\noindent
```

```
\textbf{sentence} a group of words terminated by  
a period, exclamation point, or question mark.
```

```
\setlength{\hangindent}{30pt}
```

```
\noindent
```

```
\textbf{paragraph} a group of sentences terminated by a  
blank line or by the new paragraph command.
```

produces

┌

sentence a group of words terminated by a period, exclamation point, or question mark.

paragraph a group of sentences terminated by a blank line or by the new paragraph command.

└

Notice that the `\setlength` command must be repeated for each paragraph.

Sometimes you may want to change the value of `\hangafter`, the length command that specifies the number of lines not to be indented. The default value is 1. To change it to 2, use the command

```
\setlength{\hangafter}{2}
```

For more about the `\setlength` command; see Section 15.5.2. LC3 discusses in Section 3.1.4 the style parameters of a paragraph.

The preferred way to shape a paragraph or series of paragraphs is with a custom list environment (see Section 15.6).

3.6.3 Pages

There are two page breaking commands:

- `\newpage`, which breaks the page at the point of insertion but does not stretch the content;
- `\pagebreak`, which breaks the page at the point of insertion and stretches the page's content to normal length.

Text following either command starts at the beginning of the next page, indented.

As you can see, the page breaking commands are analogous to the line breaking commands discussed in Section 3.6.1. This analogy continues with the optional argument, 0 to 4:

```
\pagebreak[0] to \pagebreak[4]
\nopagebreak[0] to \nopagebreak[4]
```

There are also special commands for allowing or forbidding page breaks in multiline math displays (see Section 9.9).

When preparing the final version of a document (see Section 19.4), you may have to extend or shrink a page by a line or two to prevent it from breaking at an unsuitable line. You can do so with the `\enlargethispage` command. For instance,

```
\enlargethispage{\baselineskip}
```

adds one line to the page length. On the other hand,

```
\enlargethispage{-\baselineskip}
```

makes the page one line shorter.

```
\enlargethispage{10000pt}
```

makes the page very long.

The `*-ed` version, `\enlargethispage*`, squeezes the page as much as possible.

There are two more variants of the `\newpage` command. The

```
\clearpage
```

command does a `\newpage` and typesets all the figures and tables waiting to be processed (see Section 10.4.3). The variant

`\cleardoublepage`

is used with the `twoside` document class option (see Section 11.5). It issues the command `\clearpage` and in addition makes the next printed page a right-hand (that is, odd-numbered) page, by inserting a blank page if necessary. If for your document class this does not work, use the package `cleardoublepage.sty` in the `samples` folder.

Section 19.5 discusses the use of some of these commands in the final preparation of books.

3.6.4 Multicolumn printing

Many document classes provide the `twocolumn` option for two-column typesetting (see Section 11.5). In addition, there is a `\twocolumn` command which starts a new page by issuing a `\clearpage` and then typesets in two columns. An optional argument provides a two-column wide title. Use the `\onecolumn` command to switch back to a one-column format.

Frank Mittelbach's `multicol` package provides the more sophisticated environment `multicols`, which can start in the middle of a page, can handle more than two columns, and can be customized in a number of ways (see Section 4.3.4 of LC3).

3.7 Spaces

The judicious use of horizontal and vertical space is an important part of the formatting of a document. Fortunately, most of the spacing decisions are made by the document class, but \LaTeX has a large number of commands that allow the user to insert horizontal and vertical spacing.

Remember that \LaTeX ignores excess spaces, tabs, and end-of-line characters. If you need to add horizontal or vertical space, then you must choose from the commands in this section. Use them sparingly (see Section 2.6).

3.7.1 Horizontal spaces

In this section, we discuss fixed length horizontal space commands. Variable length horizontal space is discussed in Section 3.7.4.

When typing text, there are three commands that are often used to create horizontal space, shown between the bars in the display below:

<code>_</code>	<code>_</code>
<code>\quad</code>	<code>\quad</code>
<code>\qquad</code>	<code>\qquad</code>

The `\quad` command creates a 1 em space and `\qquad` creates a 2 em space (see Section 3.7.3). The interword space created by `_` can both stretch and shrink. There are other commands that create smaller amounts of space. All the math spacing commands

of Section 8.1.3—with the exception of `\mspace`—can be used in ordinary text (see Appendixes A.8 and B.6), but the `\hspace` and `\phantom` commands may be more appropriate.

The `\hspace` command takes a length as a parameter. The length may be negative. For example,

```
\textbar\hspace{12pt}! | !
\textbar\hspace{.5in}! | !
\textbar\hspace{1.5cm}! | !
!\hspace{-.5cm}\textbar ! |
```

The command `\hspace` is often used with a negative argument for illustrations.

The `` command produces a space the width and height of the space that would be occupied by its typeset argument

```
\textbar need space\textbar |need space|
\textbar\phantom{need space}\textbar | |
```

and

```
alpha \phantom{beta} gamma \phantom{delta}\\
\phantom{alpha} beta \phantom{gamma} delta
```

produces

```
┌
alpha      gamma
      beta      delta
└
```

The `\phantom` command is very useful for fine-tuning aligned math formulas (see Sections 8.1 and 9.5.3). The variant

```
\hphantom{argument}
```

creates a space with the horizontal dimension that would be occupied by its typeset *argument* and with zero height.

For instance, the last two lines of the dedication of this book were typed as follows in the `flushright` environment:

```
\textbf{Kate} (8), \phantom{\textbf{Jay} (3)}\\[8pt]
\textbf{Jay} (3)
```

See Appendix B.6 for a table of all horizontal text-spacing commands.

It is easy to see how we can emulate the `\pause` command of `beamer`—see Section 13.2.1—with the `\phantom` command.

Horizontal space variant

When \LaTeX typesets a line, it removes all spaces from the beginning of the line, including the space created by `\hspace`, `\quad`, and other spacing commands. Using the *-ed variant of `\hspace`, `\hspace*`, prevents \LaTeX from removing the space you have specified.

For example,

```
And text\\
\hspace{20pt}And text\\
\hspace*{20pt}And text
```

is typeset as

```
┌
And text
And text
└   And text
```

Use the `\hspace*` command for creating customized indentation. To indent a paragraph by 24 points, give the command

```
\noindent\hspace*{24pt}And text
```

which typesets as

```
┌
   And text
└
```

To break a line and indent the next line by 24 points, give the command

```
And text\\
\hspace*{24pt}And text
```

which produces

```
┌
And text
└   And text
```

3.7.2 Vertical spaces

You can add some interline space with the command `\length` as discussed in Section 3.6.1. You can also do it with the `\vspace` command, which works just like the `\hspace` command (see Section 3.7.1), except that it creates vertical space. Here are some examples:

```
\vspace{12pt} \vspace{.5in} \vspace{1.5cm}.
```

Standard amounts of vertical space are provided by the three commands

```
\smallskip \medskip \bigskip
```

The space these commands create depends on the document class and the font size. With the document class and font I am using for this book, they represent a vertical space of 3 points, 6 points, and 12 points, respectively. 12 points is the baseline skip in standard L^AT_EX documents with the default 10pt option.



Rule: Vertical space commands

All vertical space commands add the vertical space *after* the typeset line in which the command appears.

To obtain

```
┌
end of text.
```

```
└      New paragraph after vertical space
```

```
type
end of text.
```

```
\vspace{12pt}
```

New paragraph after vertical space

The following example demonstrates how the vertical space is unexpectedly positioned when the command creating it does not start a new paragraph:

```
end of text.
\vspace{12pt}
```

The following example illustrates the unexpected way the vertical space is placed if the command that creates it does not start a new paragraph:

It typesets as

```
┌
end of text. The following example illustrates the unexpected way the vertical
└      space is placed if the command that creates it does not start a new paragraph:
```


Vertical space variants

L^AT_EX removes vertical space from the beginning and end of each page, including space produced by `\vspace`. The space created by the variant `\vspace*` is not removed by L^AT_EX under any circumstances. Use this command, for instance, to start the typeset text (say, of a letter) not at the top of the page.

The `\phantom` command has also a vertical variant: `\vphantom`. The command `\vphantom{argument}` creates a vertical space with the vertical dimension that would be occupied by its typeset argument, *argument*.

3.7.3 Relative spaces

The length of a space is usually given in *absolute units*: 12pt (points), .5cm (centimeters), 1.5in (inches). Sometimes, *relative units*, em and ex, are more appropriate, units that are relative to the size of the letters in the current font. The unit 1 em is approximately the width of an M in the current font, while 1 ex is approximately the height of an x in the current font. These units are used in commands such as

`\hspace{12em}` and `\vspace{12ex}`

The `\quad` and `\qquad` commands (Section 3.7.1) produce 1 em and 2 em spaces.

3.7.4 Expanding spaces**Horizontal spaces**

The `\hfill`, `\dotfill`, and `\hrulefill` commands fill all available space in the line with spaces, dots, or a horizontal line, respectively. If there are two of these commands on the same line, the space is divided equally between them. These commands can be used to center text, to fill lines with dots in a table of contents, and so on.

To obtain

┌	2. Boxes.....	34
	ABC and DEF	
└	ABC and DEF	

type

```
2. Boxes\dotfill 34\\
ABC\hfill and\hfill DEF\\
ABC\hrulefill and\hrulefill DEF
```

In a centered environment—such as a `\title` (see Section 11.2.1) or a `center` environment (see Section 4.3)—you can use `\hfill` to set a line flush right:

```

┌
                                     This is the title
                                     First Draft
                                     Author
└

```

To achieve this effect, type

```

\begin{center}
  This is the title\\
  \hfill First Draft\\
  Author
\end{center}

```

Vertical spaces

The vertical analogue of `\hfill` is `\vfill`. This command fills the page with vertical space so that the text before the command and the text after the command stretch to the upper and lower margin. You can play the same games with it as with `\hfill` in Section 3.7.4.

The command `\vfill` stands for `\vspace{\fill}`, so it is ignored at the beginning of a page. Use `\vspace*{\fill}` if you need it at the beginning of a page.

3.8 Boxes

Sometimes it can be useful to typeset text in an imaginary box, and treat that box as a single large character. A single-line box can be created with the `\mbox` or `\makebox` commands and a multiline box of a prescribed width can be created with the `\parbox` command or `minipage` environment.

3.8.1 Line boxes

The `\mbox` command provides a *line box* that typesets its argument without line breaks. As a result, you may find the argument extending into the margin. The resulting box is handled by \LaTeX as if it were a single large character. For instance,

```
\mbox{database}
```

causes \LaTeX to treat the eight characters of the word `database` as if they were one. This technique has a number of uses. It prevents \LaTeX from breaking the argument

(see Section 3.4.9). It also allows you to use the phrase in the argument in a formula (see Section 7.4.6).

The argument of `\mbox` is typeset in a size appropriate for its use, for example, as a subscript or superscript. See Section 7.4.6 for an example.

Line boxes—a refinement

The `\mbox` command is the short form of the `\makebox` command. Both `\mbox` and `\text` prevent breaking the argument, but `\mbox` does not change size in subscripts and superscripts.

The full form of the `\makebox` command is

```
\makebox[width][alignment]{text}
```

where the arguments are:

- *width*, the (optional) width of the box. If *[width]* is omitted, the box is as wide as necessary to enclose its contents.
- *alignment*, (optionally) one of *c* (the default), *l*, *r*, or *s*. The text is centered by default, *l* sets the argument flush left, *r* right, and *s* stretches the text the full length of the box if there is blank space in the argument.
- *text*, the text in the box.

A *width* argument can be specified in inches (in), centimeters (cm), points (pt), em, or ex (see Sections 3.7.3 and 15.5.2).

The following examples,

```
\makebox{Short title.}End\\
\makebox[2in][l]{Short title.}End\\
\makebox[2in]{Short title.}End\\
\makebox[2in][r]{Short title.}End\\
\makebox[2in][s]{Short title.}End
```

typeset as

```
┌
Short title.End
Short title.           End
      Short title.      End
                Short title.End
└ Short               title.End
```

The optional width argument, *width*, can use four length commands:

```
\height \depth \totalheight \width
```

These are the dimensions of the box that would be produced without the optional width argument.

Here is a simple example. The command

```
\makebox{hello}
```

makes a box of width `\width`. To typeset `hello` in a box three times the width, that is, in a box of width: `3\width`, use the command

```
\makebox[3\width]{hello}
```

So

```
start\makebox[3\width]{hello}end
```

typesets as

```
┌
start    hello    end
└
```

The formal definition of these four length commands is the following:

- `\height` is the height of the box above the baseline;
- `\depth` is the depth of the box below the baseline;
- `\totalheight` is the sum of `\height` and `\depth`;
- `\width` is the width of the box.

Line boxes—more examples

These constructs are important enough to be illustrated with some more examples in the `parbox.tex` sample file:

```
\documentclass{article}
\begin{document}
```

```
\subsubsection*{Example 1. Basic usage}
```

```
\parbox{5cm}{This is a simple parbox
containing multiple lines of text
to demonstrate the basic usage
of the \textbackslash parbox command in \LaTeX.}
```

```
\subsubsection*{Example 2. Aligning parboxes}
```

```

\parbox[t]{5cm}{This is the first parbox.}
\hspace{1cm}
\parbox[c]{5cm}{This is the second parbox,
center-aligned to the first one.}

\subsubsection*{Example 3. Specifying height and inner alignment}

\parbox[c][5cm][t]{5cm}{This parbox has a specified height of 5cm,
and the text inside is aligned to the top.}
\hspace{1cm}
\parbox[c][5cm][c]{5cm}{This parbox has a specified height of 5cm,
and the text inside is centered.}

\end{document}

```

Figure 3.1 shows the typeset sample file.

Example 1. Basic usage

This is a simple parbox containing multiple lines of text to demonstrate the basic usage of the `\parbox` command in \LaTeX .

Example 2. Aligning parboxes

This is the first parbox.

This is the second parbox,
center-aligned to the first one.

Example 3. Specifying height and inner alignment

This parbox has a specified height of 5cm, and the text inside is aligned to the top.

This parbox has a specified height of 5cm, and the text inside is centered.

Figure 3.1: More uses of `\parbox`

Line boxes—a variant

There is an interesting variant of the `\makebox` command. The `\rlap` command makes a box and pretends that it is of width zero. For instance,

```
\newcommand{\circwithdot}
{\mathbin{\rlap{$\mspace{2mu}\cdot$}\hbox{$\circ$}}}
```

defines the command `\circwithdot`, so you can type

```
$f\circwithdot\varphi$
```

which prints as $f \circ \varphi$. There is also an `\llap` command.

3.8.2 Frame boxes

Boxed text is very emphatic. For example, Do not touch! is typed as

```
\fbox{Do not touch!}
```

This is a *frame box*, hence the command `\fbox` or `\framebox`.

Boxed text cannot be broken, so if you want a frame around more than one line of text, you should put the text as the argument of a `\parbox` command or within a `minipage` environment (see Section 3.8.3), and then put that into the argument of an `\fbox` command. For instance,

```
\fbox{\parbox{3in}{Boxed text cannot be broken,
so if you want to frame more than one line
of text, place it in the argument of a
\bsl\texttt{parbox}
command or within a
\texttt{minipage} environment.}}
```

produces

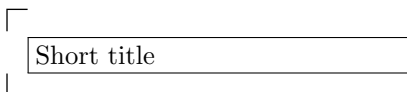
Boxed text cannot be broken, so if you want to frame more than one line of text, place it in the argument of a `\parbox` command or within a `minipage` environment.

The `\bsl` command is defined in Section 15.1.1. See Section 8.7 for boxed formulas.

The `\framebox` command works exactly like `\makebox`, except that it draws a frame around the box.

```
\framebox[2in][l]{Short title}
```

produces



You can use this command to typeset the number 1 in a square box, as required by the title of Michael Doob's [8]:

T_EX Starting from 1

`\framebox{\makebox[\totalheight]{1}}`

which typesets as



Note that

`\framebox[\totalheight]{1}`

typesets as



which is not a square box. Indeed, `\totalheight` is the height of 1, which becomes the width of the box. The total height of the box, however, is the height of the character 1 to which you have to add twice the `\fboxsep`, the separation between the contents of the box and the frame, defined as 3 points, and twice the `\fboxrule`, the width of the line, or rule, defined as 0.4 points. These lengths are in general also added to the width of the box, but not in this case, because we forced the width to equal the height of the character.

You can use the `\fbox` command to frame the name of an author:

`\author{\fbox{author's name}}`

3.8.3 Paragraph boxes

A paragraph box works like a paragraph. The text it contains is wrapped around into lines. The width of these lines is set by the user.

The `\parbox` command typesets the contents of its second argument as a paragraph with a line width supplied as the first argument. The resulting box is handled by L^AT_EX as a single large character. For example, to create a 3-inch wide column,

Fred Wehrung's new result shows the limitation of
E. T. Schmidt's construction, especially for large
lattices.

type

```
\parbox{3in}{Fred Wehrung's new result shows the
limitation of E.\,T. Schmidt's construction,
especially for large lattices.}
```

Paragraph boxes are especially useful when working within a tabular environment. See the subsection on refinements in Section 4.6 for examples of multiline entries.

The width of the paragraph box can be specified in inches (in), centimeters (cm), points (pt), or the relative measurements em and ex (see Section 3.7.3), among others (see Section 15.5.2 for a complete listing of measurement units).



Rule: The `\parbox` command requires two arguments. Dropping the first argument results in a message such as

```
! Missing number, treated as zero.
<to be read again>
T
1.175
```

Dropping the second argument does not yield a message but the result is probably not what you intended. The next character is taken as the contents of the `\parbox`.

Paragraph box refinements

The “character” created by a `\parbox` command is placed on the line so that its vertical center is aligned with the center of the line. An optional first argument `b` or `t` forces the paragraph box to align along its bottom or top. For an example; see Section 4.6. The full syntax of `\parbox` is

```
\parbox[alignment][height][inner-alignment]{width}{text}
```

Just as for the `\makebox` command (see Section 3.8.1), the

```
\height \depth \totalheight and \width
```

commands may be used in the *height* argument instead of a numeric argument.

The *inner-alignment* argument is the vertical equivalent of the *alignment* argument for `\makebox`, determining the position of *text* within the box and it may be any one of `t`, `b`, `c`, or `s`, denoting top, bottom, centered, or stretched alignment, respectively. When the *inner-alignment* argument is not specified, it defaults to *alignment*.

Paragraph box as an environment

The `minipage` environment is very similar to the `\parbox` command. It typesets the text in its body using a line width supplied as an argument. It has an optional argument for bottom or top alignment, and the other `\parbox` refinements also apply. The difference is that the `minipage` environment can contain displayed text environments discussed in Chapter 4.

The `minipage` environment can also contain footnotes (see Section 3.5.2) that are displayed within the `minipage`; see Section 6.8.1 of LC3.

3.8.4 Marginal comments

Do not
use this
often

A variant of the paragraph box, the `\marginpar` command, allows you to add marginal comments. So

```
\marginpar{Do not use this often}
```

produces the comment displayed in the margin.

Ⓐ

The AMS warning in the book [18] (also displayed here below the marginal comment) is defined as

```
\marginpar{{\Large%  
\textcircled{\raisebox{.7pt}{\normalsize\textbf A}}}}
```

The `\textcircled` command is discussed in Section 3.4.7, while the `\raisebox` command is introduced in Section 3.8.6.



Rule: Marginal comments and math environments

Do not use marginal comments in equations or multiline math environments.



Tip Avoid using too many marginal comments on any given page— \LaTeX may have to place some of them on the next page.

If the document is typeset two-sided, then the marginal comments are set in the outside margin. The form

```
\marginpar[left-comment]{right-comment}
```

uses the required argument *right-comment* when the marginal comment is set in the right margin and the optional argument *left-comment* when the marginal comment is set in the left margin.

The width of the paragraph box for marginal comments is stored in the length command `\marginparwidth` (see Section 15.5.2 for length commands). If you want to change it, use

```
\setlength{\marginparwidth}{new_width}
```

as in

```
\setlength{\marginparwidth}{90pt}
```

The default value of this width is set by the document class. If you want to know the present setting, type

```
\the\marginparwidth
```

in your document and typeset it, or, in interactive mode (see Section 15.1.8), type

```
*\showthe\marginparwidth
```

(* is the interactive prompt).

See Section 3.5.11 of LC3 for other style parameters pertaining to marginal notes.

3.8.5 Solid boxes

A solid filled box is created with a `\myrule` command. The first argument is the width and the second is the height. For instance, to obtain

```
┌
└ end of proof symbol: ■
```

type

```
end of proof symbol: \rule{1.6ex}{1.6ex}
```

In fact, this symbol is usually slightly lowered:

```
┌
└ end of proof symbol: ■
```

This positioning is done with an optional first argument:

```
end of proof symbol: \rule[-.23ex]{1.6ex}{1.6ex}
```

Here is an example combining `\myrule` with `\makebox` and `\hrulefill`:

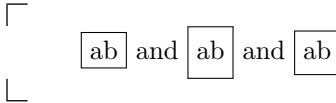
```
1 inch:\quad\makebox[1in]{\rule{.4pt}{4pt}%
    \hrulefill\rule{.4pt}{4pt}}
```

which produces

```
┌
└ 1 inch: ───────────
```

Struts

Solid boxes of zero width are called *struts*. Struts are invisible, but they force L^AT_EX to make room for them, changing the vertical alignment of lines. Standard struts can also be added with the `\strut` or `\mathstrut` command. To see how struts work, compare



typed as

`\fbox{ab}` and `\fbox{\strut ab}` and `\fbox{\mathstrutab}`

Struts are especially useful for fine tuning tables (see Section 4.6) and formulas (see math struts in Section 8.5).

**Rule: Zero distance**

Use `0pt`, `0in`, `0cm`, `0em` for zero width, `0` by itself is not acceptable.

For example, `\rule{0}{1.6ex}` gives the message

```
! Illegal unit of measure (pt inserted).
<to be read again>
          h
1.251 \rule{0}{1.6ex}
```

If the `\rule` command has no argument or only one, L^AT_EX generates a message. For instance, `\rule{1.6ex}` gives the message

```
! Paragraph ended before \@rule was complete.
```

or

```
! Missing number, treated as zero.
```

In the first message, the reference to `\@rule` suggests that the problem is with the `\rule` command. Checking the syntax of the `\rule` command, you find that an argument is missing. The second message is more informative, since there is, indeed, a missing number.

3.8.6 Fine tuning boxes

The command

`\raisebox{displacement}{text}`

typesets *text* in a box with a vertical *displacement*. If *displacement* is positive, the box is raised; if it is negative, the box is lowered.

The `\raisebox` command allows us to play games:

```
fine-\raisebox{.5ex}{tun}\raisebox{-.5ex}{ing}
```

produces fine-tun_{ing}.

The `\raisebox` command has two optional arguments:

```
\raisebox{0ex}[1.5ex][0.75ex]{text}
```

forces L^AT_EX to typeset *text* as if it extended 1.5 ex above and 0.75 ex below the line, resulting in a change in the interline space above and below the line. A simple version of this command, `\smash`, is discussed in Section 8.5.

In the AMS warning in the book [18] (shown on page 93), the `\raisebox` command is used to properly center the bold A in the circle:

```
\Large\textcircled{\raisebox{.7pt}{\normalsize\textbf{A}}}
```



Text environments

There are three types of text environments in \LaTeX :

1. Displayed text environments; text within such an environment usually is typeset with some vertical space around it
2. Text environments that create a “large symbol”
3. Style and size environments

We start by discussing a very important rule about blank lines in displayed text environments. Then we proceed in Section 4.2 to the most often used displayed text environments: lists. We continue with the style and size environments in Section 4.3.

The most important displayed text environments in math are proclamations or theorem-like structures, proclamations with style, and the proof environment, discussed in detail in Sections 4.4 and 4.5.

The `tabular` environment discussed in Section 4.6 produces a “large symbol,” a table, which is of limited use in math. In Section 4.7, we discuss the `tabbing` environment, which may be used for computer code. The legacy environments `quote`,

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-55281-6_4

quotation, and `verse` are discussed in Section 4.8, along with the `verbatim` environment, which is often used to display \LaTeX source in a typeset \LaTeX document.

4.1 Some general rules

As you know, blank lines play a special role in \LaTeX , usually indicating a paragraph break. Since displayed text environments structure the printed display themselves, the rules about blank lines are relaxed somewhat. However, a blank line trailing an environment signifies a new paragraph for the text following the environment.



Rule: Blank lines in displayed text environments

1. Blank lines are ignored immediately after `\begin{name}` or immediately before `\end{name}` except in a `verbatim` environment.
2. A blank line after `\end{name}` forces the text that follows to start a new paragraph.
3. As a rule, you should not have a blank line before `\begin{name}`.
4. The line after any theorem or proof always begins a new paragraph, even if there is no blank line or `\par` command.

The page breaking commands in Section 3.6.3 apply to text environments, as does the line breaking command `\` discussed in Section 3.6.1.

4.2 List environments

\LaTeX provides three list environments: `enumerate`, `itemize`, and `description`. \LaTeX also provides a generic `list` environment that can be customized to fit your needs. See Section 15.6 on custom lists.

Most document classes redefine the spacing and some stylistic details of lists, especially since the list environments in legacy document classes are not very pleasing. In this section, the list environments are formatted as they are by our standard document class, `amsart`. Throughout the rest of the book, lists are formatted as specified by this book's designer.

4.2.1 Numbered lists

A *numbered list* is created with the `enumerate` environment:

┌ This space has the following properties:

- (1) Grade 2 Cantor;
- (2) Half-smooth Hausdorff;
- (3) Metrizable smooth.

└ Therefore, we can apply the Main Theorem.

typed as

```
\noindent This space has the following properties:
\begin{enumerate}
  \item Grade 2 Cantor\label{Cantor};
  \item Half-smooth Hausdorff\label{Hausdorff};
  \item Metrizable smooth\label{smooth}.
\end{enumerate}
Therefore, we can apply the Main Theorem.
```

Each item is introduced with an `\item` command. The numbers \LaTeX generates can be labeled and cross-referenced (Section 10.4.2). This construct can be used in theorems and definitions, for listing conditions or conclusions.

If you use `\item` in the form `\item[]`, you get an unnumbered item in the list, while `\item[a]` replaces the number of the item with *a*. This is another form of absolute referencing, see Section 10.4.2.



Rule: Do not label absolute references. It may lead to problems that are hard to explain.

4.2.2 Bulleted lists

A *bulleted list* is created with the `itemize` environment:

┌ We set out to accomplish a variety of goals:

- To introduce the concept of smooth functions.
- To show their usefulness in differentiation.
- To point out the efficacy of using smooth functions in Calculus.

└

is typed as

```
\noindent We set out to accomplish a variety of goals:
\begin{itemize}
  \item To introduce the concept of smooth functions.
  \item To show their usefulness in differentiation.
  \item To point out the efficacy of using smooth
        functions in Calculus.
\end{itemize}
```

4.2.3 Captioned lists

In a *captioned list* each item has a title (caption) specified by the optional argument of the `\item` command. Such lists are created with the `description` environment:

```
┌
In this introduction, we describe the basic techniques:
  Chopped lattice: a reduced form of a lattice;
  Boolean triples: a powerful lattice construction;
  Cubic extension: a subdirect power flattening the congruences.
└
```

is typed as

```
\noindent In this introduction, we describe
  the basic techniques:
\begin{description}
  \item[Chopped lattice] a reduced form of a lattice;
  \item[Boolean triples] a powerful lattice construction;
  \item[Cubic extensions] a subdirect power flattening
    the congruences.
\end{description}
```

4.2.4 A rule and combinations

There is only one rule you must remember.



Rule: List environments

An `\item` command must immediately follow `\begin{enumerate}`, `\begin{itemize}`, or `\begin{description}`.

Of course, spaces and line breaks can separate them.

If you break this rule, you get a message. For instance,

```
\begin{description}
This is wrong!
  \item[Chopped lattice] a reduced lattice;
```

gives the message

```
! LaTeX Error: Something's wrong--perhaps a missing \item.
```

```
1.105  \item[Chopped lattice]
          a reduced lattice;
```


If you see this message, remember the rule for list environments and check for text preceding the first `\item`.

You can nest up to four list environments; for instance,

- ┌
- (1) First item of Level 1.
 - (a) First item of Level 2.
 - (i) First item of Level 3.
 - (A) First item of Level 4.
 - (B) Second item of Level 4.
 - (ii) Second item of Level 3.
 - (b) Second item of Level 2.
 - (2) Second item of Level 1.

Referencing the second item of Level 4: 1(a)iB

└

which is typed as

```
\begin{enumerate}
  \item First item of Level 1.
  \begin{enumerate}
    \item First item of Level 2.
    \begin{enumerate}
      \item First item of Level 3.
      \begin{enumerate}
        \item First item of Level 4.
        \item Second item of Level 4.\label{level4}
      \end{enumerate}
      \item Second item of Level 3.
    \end{enumerate}
    \item Second item of Level 2.
  \end{enumerate}
  \item Second item of Level 1.
\end{enumerate}
```

Referencing the second item of Level 4: `\ref{level4}`

Note that the label `level4` collected all four of the counters (see Section [10.4.2](#)).

You can also mix list environments:

- ┌
- (1) First item of Level 1.
 - First item of Level 2.
 - (a) First item of Level 3.
 - First item of Level 4.
 - Second item of Level 4.
 - (b) Second item of Level 3.
 - Second item of Level 2.
 - (2) Second item of Level 1.

Referencing the second item of Level 4: 1a

└

which is typed as

```
\begin{enumerate}
  \item First item of Level 1.
  \begin{itemize}
    \item First item of Level 2.
    \begin{enumerate}
      \item First item of Level 3.
      \begin{itemize}
        \item First item of Level 4.
        \item Second item of Level 4.\label{enums}
      \end{itemize}
      \item Second item of Level 3.
    \end{enumerate}
    \item Second item of Level 2.
  \end{itemize}
  \item Second item of Level 1.
\end{enumerate}
Referencing the second item of Level 4: \ref{enums}
```

Now the label `enums` collects only the two `enumerate` counters (see Section 10.4.2).

The indentations are, of course, not needed. I use them to keep track of the level of nesting.

In all three types of list environment, the `\item` command may be followed by an optional argument, which is displayed at the beginning of the typeset item:

```
\item[label]
```

Note that for `enumerate` and `itemize` the resulting typography may leave something to be desired.



Rule: If the text following an `\item` command starts with an opening square bracket, `[`, then \LaTeX thinks that `\item` has an optional argument. To prevent this problem from occurring, type `[` as `{[}`. Similarly, a closing square bracket, `]`, *inside* the optional argument should be typed as `{]}`.



Tip You may want to use a list environment solely for the way the items are displayed, without any labels. You can achieve this by using `\item[]`.

You can change the style of the numbers in an `enumerate` environment by re-defining the counter as suggested in Section 15.5.1:

```
\renewcommand{\labelenumi}{\normalfont (\roman{enumi})}
```

The labels then are displayed as (i), (ii), and so on. This modification only works if you do not want to reference these items. If you want the `\ref` command to work properly, use David Carlisle's `enumerate` package, which changes the labeling in lists.

Here's a basic usage of the `enumerate` package:

```
\begin{enumerate}[I.]
  \item First item
  \item Second item
  \item Third item
\end{enumerate}
```

typesets as

```
I. First item
II. Second item
III. Third item
```

In this example, the optional argument of the `enumerate` environment (`I.`) instructs \LaTeX to label the items with capital roman numerals followed by a period.

There are many options available:

- A Labels with capital letters.
- a Labels with lowercase letters.
- I Labels with uppercase roman numerals.
- i Labels with lowercase roman numerals.
- 1 Labels with numbers (this is the default if no option is given).

For each option X , there is an option $X.$ that adds a period.

See also Section 15.2.1. Section 3.3 of LC3 goes very deeply into the list environments. For custom lists; see Section 15.6.

4.3 Style and size environments

There are several text environments that allow you to set font characteristics. They have the same names as their corresponding command declarations:

```
rmfamily sffamily ttfamily
upshape itshape em slshape scshape
bfseries
```

For instance,

```
\begin{ttfamily}
  text
\end{ttfamily}
```

typesets *text* just like `{\ttfamily text}` would. Remember to use the command-declaration names for the environment names, that is, use `rmfamily`, not `textrm` and `ttfamily`, not `texttt` (see 5.2). There are also text environments for changing the font size, from *tiny* to *Huge* (see Section 5.4).

If you are getting overwhelmed by the large number of environments changing style and size, consult Tables 5.1 and 5.2 (see also Appendix B.3.2).

Horizontal alignment of a paragraph is controlled by the `flushleft`, `flushright`, and `center` environments. Within the `flushright` and `center` environments, it is customary to force new lines with the `\\` command, while in the `flushleft` environment, you normally allow \LaTeX to wrap the lines.

These text environments can be used separately or in combination, as in

```
┌
    The simplest text environments set the printing style and size.
    The commands and the environments have similar names.
```

```
└
```

typed as

```
\begin{flushright}
  The \begin{bfseries}simplest\end{bfseries}
  text environments set the
  printing style and size.\\
  The commands and the environments have similar names.
\end{flushright}
```

There are command declarations that correspond to these environments:

- `\centering` centers text,
- `\raggedright` left aligns text,
- `\raggedleft` right aligns text.

The effect of one of these commands is almost the same as that of the corresponding environment except that the environment places additional vertical space before and after the displayed paragraphs. For such a command declaration to affect the way a paragraph is formatted, the scope must include the whole paragraph, including the blank line at the end of the paragraph, preferably indicated with a `\par` command.

The `\centering` command is used often with the `\includegraphics` command (see Section 10.4.3).

4.4 Proclamations (theorem-like structures)

Theorems, lemmas, definitions, and so forth are a major part of mathematical writing. In \LaTeX , these constructs are typed in displayed text environments called *proclamations* or *theorem-like structures*.

In the `art1.tex` sample article (see p. 4), there is only a single theorem.

In the `art2.tex` sample article (see pp. 299–302), there are a number of different proclamations in a variety of styles, with varying degrees of emphasis. Proclamations with style are discussed in Section 4.4.2.

There are two steps required to use a proclamation:

Step 1 Define the proclamation with a `\newtheorem` command *in the preamble* of the document. For instance, the line

```
\newtheorem{theorem}{Theorem}
```

defines a `theorem` environment.

Step 2 Invoke the proclamation as an environment *in the body* of your document. Using the proclamation definition from Step 1, type

```
\begin{theorem}
  My first theorem.
\end{theorem}
```

to produce a theorem:

```
┌
└ Theorem 1. My first theorem.
```

In the proclamation definition

```
\newtheorem{theorem}{Theorem}
```

the first argument, `theorem`, is the name of the environment that invokes the theorem. The second argument, `Theorem`, is the name that is used when the proclamation is typeset. \LaTeX numbers the theorems automatically and typesets them with vertical space above and below. The phrase **Theorem 1.** appears, followed by the theorem itself, which may be emphasized. Of course, the formatting of the theorem depends on the document class and the proclamation style (see Section 4.4.2).

You may also specify an optional argument,

```
\begin{theorem}[The Fuchs-Schmidt Theorem]
  The statement of the theorem.
\end{theorem}
```

that appears as the name of the theorem:

┌ **Theorem 1** (The Fuchs-Schmidt Theorem). *The statement of the theorem.*
└

\LaTeX is very fussy about how proclamations are defined. For example, in the introductory article `art1.tex` (see Section 1.8), if the closing brace is dropped from the end of line 4,

```
\newtheorem{theorem}{Theorem
you get a message such as

Runaway argument?
{Theorem
./art1.tex:5: Paragraph ended before \@ynthm was complete.
<to be read again>

\par
```

1.5

Line 5 is the line after the `\newtheorem` commands. The message conveys the information that something is wrong in the paragraph before line 5.



Rule: Lists in proclamations

If a proclamation starts with a list environment, precede the list by `\hfill`.

If you do not, as in

```
\begin{definition}\label{D:prime}
  \begin{enumerate}
    \item  $u$  is if  $u = x^2$ . \label{mi1}
    \item  $u$  is if  $u = \sqrt{x}$ . \label{mi2}
  \end{enumerate}
\end{definition}
```

your typeset list starts on the first line of the proclamation:

```

┌
Definition 1.      (1) u is bold if  $u = x^2$ .
                    (2) u is thin if  $u = \sqrt{x}$ .
└

```

If you add the `\hfill` command,

```

\begin{definition}\hfill
\begin{enumerate}

```

the list in the definition typesets correctly:

```

┌
Definition 1.
    (1) u is bold if  $u = x^2$ .
    (2) u is thin if  $u = \sqrt{x}$ .
└

```

Consecutive numbering

If you want to number two sets of proclamations consecutively, you can do so by first defining one proclamation, and then using its name as an optional argument of the second proclamation. For example, to number the lemmas and propositions in your paper consecutively, you type the following two lines in your preamble:

```

\newtheorem{lemma}{Lemma}
\newtheorem{proposition}[lemma]{Proposition}

```

Lemmas and propositions are then consecutively numbered as **Lemma 1**, **Proposition 2**, **Proposition 3**, and so on.

Let me emphasize: The optional argument of a proclamation definition must be the name of a proclamation that *has already been defined*. If you violate this rule, and type

```

\newtheorem{proposition}[lemma]{Proposition}
\newtheorem{lemma}{Lemma}

```

you get an error message:

```
LaTeX Error: No counter 'lemma' defined.
```

Numbering within a section

The `\newtheorem` command may also have a different optional argument; it causes L^AT_EX to number the proclamations within sections. For example,

```

\newtheorem{lemma}{Lemma}[section]

```

numbers the lemmas in Section 1 as **Lemma 1.1** and **Lemma 1.2**. In Section 2, you have **Lemma 2.1** and **Lemma 2.2**, and so on.

Instead of `section`, you may use any sectioning command provided by the document class, such as `chapter`, `section`, and `subsection`.

Consecutive numbering and numbering within a section can be combined. For example,

```
\newtheorem{lemma}{Lemma}[section]
\newtheorem{proposition}[lemma]{Proposition}
```

sets up the `lemma` and `proposition` environments so that they are numbered consecutively within sections: **Lemma 1.1**, **Proposition 1.2**, **Proposition 1.3** and **Proposition 2.1**, **Lemma 2.2**, and so on.

4.4.1 The full syntax

The full form of `\newtheorem` is

```
\newtheorem{envname}[procCounter]{Name}[secCounter]
```

where the two optional arguments are mutually exclusive, and

envname is the name of the environment to be used in the body of the document.

For instance, you may use `theorem` for the *envname* of a theorem, so that a theorem is typed inside a theorem environment. Of course, *envname* is just a label; you are free to choose any environment name, such as `thm` or `george` (as long as *the name is not in use as the name of another command or environment*). This argument is also the name of the counter \LaTeX uses to number these text environments.

procCounter is an optional argument. It sets the new proclamation to use the counter of a previously defined proclamation and the two proclamations are consecutively numbered.

Name is the text that is typeset when the proclamation is invoked. So if `Theorem` is given as *Name*, then you get **Theorem 1**, **Theorem 2**, and so on in your document.

secCounter is an optional argument that causes the *Name* environments to be numbered within the appropriate sectioning units. So if `theorem` is the *envname* and `section` is the *secCounter*, then in Section 1 you have **Theorem 1.1**, **Theorem 1.2**, and so on. In Section 2 you get **Theorem 2.1**, **Theorem 2.2**, and so on. Proclamations may be numbered within subsections, sections, chapters, or any other sectioning unit automatically numbered by \LaTeX .

4.4.2 Proclamations with style

You can choose one of three styles for your proclamations by preceding the definitions with the `\theoremstyle{style}` command, where *style* is one of the following:

- `plain`, the most emphatic (wow),
- `definition`,
- `remark`, the least emphatic.

There are a few extra options, including the `\newtheorem*` command, an unnumbered version of `\newtheorem`.

The following commands set the styles in the `art2.tex` article. The typeset sample article (on pages 299–302) shows how the chosen styles affect the typeset proclamations.

```
\theoremstyle{plain}
\newtheorem{theorem}{Theorem}
\newtheorem{corollary}{Corollary}
\newtheorem*{main}{Main~Theorem}
\newtheorem{lemma}{Lemma}
\newtheorem{proposition}{Proposition}

\theoremstyle{definition}
\newtheorem{definition}{Definition}

\theoremstyle{remark}
\newtheorem*{notation}{Notation}
```

A proclamation created by a `\newtheorem` command has the style of the last `\theoremstyle` command preceding it. The default style is `plain`.

Three examples

Here are three sets of proclamation definitions to illustrate different styles and numbering schemes.

Example 1

```
\theoremstyle{plain}
\newtheorem{theorem}{Theorem}
\newtheorem{lemma}[theorem]{Lemma}
\newtheorem{definition}[theorem]{Definition}
\newtheorem{corollary}[theorem]{Corollary}
```

In a document with this set of proclamation definitions you can use theorems, lemmas, definitions, and corollaries, typeset in the most emphatic (plain) style. They are all numbered consecutively: **Definition 1**, **Definition 2**, **Theorem 3**, **Corollary 4**, **Lemma 5**, **Lemma 6**, **Theorem 7**, and so on.

Example 2

```
\theoremstyle{plain}
\newtheorem{theorem}{Theorem}
\newtheorem*{main}{Main Theorem}
\newtheorem{definition}{Definition}[section]
\newtheorem{lemma}[definition]{Lemma}

\theoremstyle{definition}
\newtheorem*{myrule}{myrule}
```

In this document you may use theorems, definitions, and lemmas in the most emphatic (plain) style, and unnumbered rules in the less emphatic (definition) style. Definitions and lemmas are numbered consecutively within sections. You may also use the unnumbered Main Theorem. So, for example, you may have **Definition 1.1**, **Definition 1.2**, **Main Theorem**, **myrule**, **Lemma 1.3**, **Lemma 2.1**, **Theorem 1**, and so on.

Example 3

```
\theoremstyle{plain}
\newtheorem{theorem}{Theorem}
\newtheorem{corollary}{Corollary}
\newtheorem*{main}{Main Theorem}
\newtheorem{lemma}{Lemma}
\newtheorem{proposition}{Proposition}

\theoremstyle{definition}
\newtheorem{definition}{Definition}

\theoremstyle{remark}
\newtheorem*{notation}{Notation}
```

With these proclamation definitions you can use theorems, corollaries, lemmas, and propositions in the most emphatic (plain) style, and an unnumbered Main Theorem. You can have definitions in the less emphatic (definition) style. All are separately numbered. So in the document you may have **Definition 1**, **Definition 2**, **Main Theorem**, **Lemma 1**, **Proposition 1**, **Lemma 2**, **Theorem 1**, **Corollary 1**, and so on. You can also have Notations which are unnumbered and typeset in the least emphatic (remark) style.

Number swapping

Proclamations can be numbered on the left, as for instance, **3.2 Theorem**. To accomplish this, type the `\swapnumbers` command before the `\newtheorem` command corresponding to the proclamation definition you want to change. This command affects all of the proclamation definitions that follow it, so the proclamation definitions in the preamble should be in two groups. The regular ones should be listed first, followed by the `\swapnumbers` command, then all the proclamations that swap numbers.

Do not swap numbers unless the publisher (or your language, for instance, Hungarian) demands it.

Custom theorem styles

You can define custom theorem styles with the `\newtheoremstyle` command. You should very seldom do this; the three theorem styles of the document class should suffice. For more detail; see [2] and Section 4.1.5 of LC3.

4.5 Proof environments

A proof is the contents of a proof environment. For instance,

```
┌
| Proof. This is a proof, delimited by the q.e.d. symbol.
└
```

typed as

```
\begin{proof}
This is a proof, delimited by the q.e.d.\ symbol.
\end{proof}
```

A proof is set off from the surrounding text with some vertical space. The end of the proof is marked with the symbol \square at the end of the line. There are a few examples of the proof environment in the `art2.tex` sample article (pages 293–302).

We start with the same rule for proofs as we have for proclamations on page 106.


Rule: Lists in proofs

If a proof starts with a list environment, precede the list by `\hfill`.

If you want to suppress the symbol at the end of a proof, give the command

```
\begin{proof}
...
\renewcommand{\qedsymbol}{}
\end{proof}
```

To suppress the end of the proof symbol in the whole article, give the

```
\renewcommand{\qedsymbol}{}{}
```

command in the preamble.

To substitute another phrase for *Proof*, such as *Necessity*, as in

```
[
Necessity. This is the proof of necessity. □
]
```

use the proof environment with an optional argument:

```
\begin{proof}[Necessity]
This is the proof of necessity.
\end{proof}
```

The optional argument may contain a reference, as in

```
\begin{proof}[Proof of Theorem~\ref{T:smooth}]
```

which might be typeset as

```
[
Proof of Theorem 5. This is the proof. □
]
```

It is easy to make the mistake of placing the optional argument after `\begin`:

```
\begin[Proof of Theorem~\ref{T:P*}]{proof}
```

You get a message

```
! LaTeX Error: Bad math environment delimiter.
```

```
1.91 \begin{equation}
\label{E:cong2}
```

which is not very helpful.

There is a problem with the placement of the q.e.d. symbol if the proof ends with a displayed formula (or a list environment). For instance,

```
\begin{proof}
Now the proof follows from the equation
\[
a^2 = b^2 + c^2.
\]
\end{proof}
```

typesets as

┌

Proof. Now the proof follows from the equation

$$a^2 = b^2 + c^2.$$

□

└

To correct the placement of the q.e.d. symbol, use the `\qedhere` command:

```
\begin{proof}
Now the proof follows from the equation
\[
a^2 = b^2 + c^2.\qedhere
\]
\end{proof}
```

which typesets as

Proof. Now the proof follows from the equation

$$a^2 = b^2 + c^2.$$

□

4.6 Tabular environments

A tabular environment creates a table that \LaTeX treats as a “large symbol”. In particular, a table cannot be broken across pages.

Here is a simple table,

Name	1	2	3
Peter	2.45	34.12	1.00
John	0.00	12.89	3.71
David	2.00	1.85	0.71

, typeset inline. This

looks awful, but it does make the point that the table is just a “large symbol”. The table is typed as

```
\begin{tabular}{| l | r | r | r | }
\hline
Name      & 1      & 2      & 3      \\ \hline
Peter     & 2.45   & 34.12  & 1.00   \\ \hline
John      & 0.00   & 12.89  & 3.71   \\ \hline
David     & 2.00   & 1.85   & 0.71   \\ \hline
\end{tabular}
```



```

. . .
120& \symbol{120} &\symbol{121}&\symbol{123}&
\symbol{123}&\symbol{124}&\symbol{125}&\symbol{126}
&\symbol{127} && \\ \hline
\end{tabular}

```

The second table is the same except that the numbers run from 128 to 255. The typeset table is shown in Section 3.4.4.



Rule: tabular environments

1. `\begin{tabular}` requires an argument consisting of a character l, r, or c, meaning left, right, or center alignment, for each column, and optionally, the | symbols. Each | indicates a vertical line in the typeset table. Spaces in the argument are ignored but can be used for readability.
2. Columns are separated by ampersands (&) and rows are separated by `\\`.
3. & absorbs spaces on either side.
4. The `\hline` command creates a horizontal rule in the typeset table. It is placed either at the beginning of the table (after the `\begin` line) or it must follow a `\\` command.
5. If you use a horizontal line to finish the table, you must separate the last row of the table from the `\hline` command with the `\\` command.
6. `\begin{tabular}` takes an optional argument, b or t, to specify the bottom or the top vertical alignment of the table with the baseline. The default is center alignment.

Remember to put the optional argument b or t in square brackets, as in

```
\begin{tabular}[b]{ | l | r | r | r | }
```

If you forget to place an `\hline` command right after `\\` in the last row, you get a message such as

```

! Misplaced \noalign.
\hline ->\noalign
          {\ifnum 0='}\fi \hrule \@height
          \arrayrulew...
1.9 ....00 & 1.85  & 0.71 \hline

```

More column-formatting commands

The required argument of the `tabular` environment may contain column-formatting commands of various types.

An *@-expression*, for instance, `@{.}`, replaces the space \LaTeX normally inserts between two columns with its argument. For example,

```
\begin{tabular}{r @{.} l}
  3&78\\
  4&261\\
  4
\end{tabular}
```

creates a table with two columns separated by a decimal point. In effect, you get a single, decimal-aligned column:

```
┌
    3.78
    4.261
    4.
└
```

This example is an illustration. You should use David Carlisle's `dcolumn` package if you need a decimal-aligned column.

The width of a column depends on the entries in the column by default. You can specify a width by using the `p` column specifier:

```
p{width}
```

For instance, if you want the first column of Table 4.1 to be 1 inch wide, then type

```
\begin{tabular}{| p{1in} | r | r | r | }\hline
  Name      & 1      & 2      & 3      \\ \hline
  Peter     & 2.45   & 34.12  & 1.00   \\ \hline
  John      & 0.00   & 12.89  & 3.71   \\ \hline
  David     & 2.00   & 1.85   & 0.71   \\ \hline
\end{tabular}
```

which typesets as

```
┌


|       |      |       |      |
|-------|------|-------|------|
| Name  | 1    | 2     | 3    |
| Peter | 2.45 | 34.12 | 1.00 |
| John  | 0.00 | 12.89 | 3.71 |
| David | 2.00 | 1.85  | 0.71 |


└
```


To center the items in the first column, precede *each* item with a `\centering` command (see Section 4.3). Note that the first column is actually somewhat over 1 inch wide, because of the extra space provided around the column boundaries.

The `p` column specifier can also be used for multiline entries.

Refinements

`\hline` draws a horizontal line the whole width of the table. `\cline{a-b}` draws a horizontal line from column *a* to column *b*. For instance,

```
\cline{1-3} or \cline{4-4}
```

Another useful command is `\multicolumn`, which is used to span more than one column, for example,

```
\multicolumn{3}{c}{\emph{absent}}
```

The first argument is the number of columns spanned by the entry, the second is the alignment (an optional vertical line designator `|` for this row only can also be included), and the third argument is the entry. Note that the entry for the spanned columns is in braces. An example is shown in Table 4.2, typed as follows:

```
\begin{table}[h!]
  \begin{center}
    \begin{tabular}{| l | r | r | r | } \hline
      Name    & 1    & 2    & 3\\ \hline
      Peter   & 2.45 & 34.12 & 1.00\\ \hline
      John    & \multicolumn{3}{c |}{\emph{absent}}\\ \hline
      David   & 2.00 & 1.85  & 0.71\\ \hline
    \end{tabular}
    \caption{Table with \bsl\texttt{\multicolumn}.}
    \label{Ta:mc}
  \end{center}
\end{table}
```

The next example, shown in Table 4.3, uses the `\multicolumn` and `\cline` commands together:

```
\begin{table}[t]
  \begin{center}
    \begin{tabular}{| c c | c | r | } \hline
      Name & Month & Week & Amount\\ \hline
      Peter & Jan.  & 1    & 1.00\\ \cline{3-4}
           &       & 2    & 12.78\\ \cline{3-4}
           &       & 3    & 0.71\\ \cline{3-4}
    \end{tabular}
  \end{center}
\end{table}
```

```

&      & 4      & 15.00\\ \cline{2-4}
& \multicolumn{2}{| l}{Total} & 29.49\\
\hline
John & Jan. & 1      & 12.01\\ \cline{3-4}
&      & 2      & 3.10\\ \cline{3-4}
&      & 3      & 10.10\\ \cline{3-4}
&      & 4      & 0.00\\ \cline{2-4}
& \multicolumn{2}{| l}{Total} & 25.21\\
\hline
\multicolumn{3}{|l}{Grand Total} & 54.70\\
\hline
\end{tabular}
\caption{Table with \bsl\texttt{\multicolumn}
and \bsl\texttt{\cline}.}\label{Ta:multicol+cline}
\end{center}
\end{table}

```

The `\parbox` command (see Section 3.8.3) can be used to produce a single multiline entry. Recall that the first argument of `\parbox` is the width of the box. A `p{}` width designator creates a column in which all entries can be multiline. As an example,

Name	1	2	3
Peter	2.45	34.12	1.00
John	<i>absent</i>		
David	2.00	1.85	0.71

Table 4.2: Table with `\multicolumn`.

Name	Month	Week	Amount
Peter	Jan.	1	1.00
		2	12.78
		3	0.71
		4	15.00
		Total	29.49
John	Jan.	1	12.01
		2	3.10
		3	10.10
		4	0.00
		Total	25.21
Grand Total			54.70

Table 4.3: Table with `\multicolumn` and `\cline`.

to replace Grand Total by Grand Total for Peter and John, type the last line as

```
\multicolumn{3}{l}{\parbox[b]{10em}{Grand Total\\
for Peter and John} } & 54.70\\ \hline
```

Note the use of the bottom alignment option (see Section 3.8.3). The last row of the modified table prints

Grand Total for Peter and John	54.70
-----------------------------------	-------

The spacing above Grand Total is not quite right. It can be adjusted with a strut (see Section 3.8.5),

```
\parbox[b]{10em}{\strut Grand Total\\
for Peter and John:}
```

which typesets as

Grand Total for Peter and John	54.70
-----------------------------------	-------

Finally, vertical spacing can be adjusted by redefining `\arraystretch`. For instance, in the table

┌

	Area	Students
5th Grade:	63.4 m ²	22
6th Grade:	62.0 m ²	19
Overall:	62.6 m ²	20

└

typed as

```
\begin{center}
\begin{tabular}{|r|c|c|}\hline
& \textbf{Area} & \textbf{Students}\\ \hline
\textbf{5th Grade}: & 63.4 m\textsuperscript{2} & 22\\ \hline
\textbf{6th Grade}: & 62.0 m\textsuperscript{2} & 19\\ \hline
\textbf{Overall}: & 62.6 m\textsuperscript{2} & 20\\ \hline
\end{tabular}
\end{center}
```

you may find that the rows are too crowded. The vertical spacing may be adjusted by adding the line

Table 4.4: Smokers and Nonsmokers, by Sex.

	Smoke	Don't Smoke	Total
Males	1,258	2,104	3,362
Females	1,194	2,752	3,946
Total	2,452	4,856	7,308

```
\renewcommand{\arraystretch}{1.25}
to the tabular environment. To limit its scope, add it after
\begin{center}
```

The adjusted table is typeset as

┌

	Area	Students
5th Grade:	63.4 m ²	22
6th Grade:	62.0 m ²	19
Overall:	62.6 m ²	20

└

In some tables, horizontal and vertical lines do not always intersect as desired. Fine control over these intersections is provided by the `hhline` package.

Chapter 6 of LC3 deals with tabular material, discussing many extensions, including multipage tables, decimal-point alignment, footnotes in tables, tables within tables, and so on.

4.6.1 Table styles

L^AT_EX can draw double horizontal and vertical lines in tables with ease. As a result, there are far too many double lines in L^AT_EX tables, resulting in cluttered and confusing tables. *The Chicago Manual of Style*, 17th edition [6] has almost 80 pages on tables. For simple tables it advocates a simple style, as shown in Table 4.4.

Notice:

- the generous space above and below the column heads, which has been achieved with the command `\rule[-8pt]{0pt}{22pt} \rule[-8pt]{0pt}{22pt}`;
- some extra space above the first line of data, which has been achieved with the command `\rule{0pt}{14pt}`;
- the columns of equal width, which has been achieved with `p{70pt}` commands,
- no vertical lines.



Tip Remember, tables designed by L^AT_EX, as a rule, look awful.

Most tables in this book have been designed by Simon Fear's `booktabs` package.

4.7 *Tabbing environments*

Although of limited use for mathematical typesetting, the `tabbing` environment can be useful for typing algorithms, computer programs, and so forth. L^AT_EX calculates the width of a column in the `tabular` environment based on the widest entry (see Section 4.6). The `tabbing` environment allows you to control the width of the columns.

The `\` command is the line separator, tab stops are set by `\=` and are remembered by L^AT_EX in the order they are given, and `\>` moves to the next tab position.

You can easily reset tab positions. For instance, if you are past the second tab position by using `\>` twice, and there is a third tab position, the `\=` command resets it.

Lines of comments may be inserted with the `\kill` command; see the examples below, or with the `%` character. The difference is that a line with `\kill` can be used to set tab stops, whereas a commented out line cannot.

A simple example:

┌

```
PrintTime
    Block[timing],
        timing = Timing[expr];
        Print[ timing[[1]] ];
    ]
End[]
```

└

typed as

```
{\ttfamily
\begin{tabbing}
    Print\=Time\\
    \>Block\=[timing,\\
    \>\>timing = Timing[expr];\\
    (careful with initialization)\kill
    \>\>Print[ timing[[1]] ];\\
    \>]\\
End[]
\end{tabbing}
}% end \ttfamily
```

An alternative method is to use a line to set the tab stops, and then `\kill` the line so it does not print:

```
{\ttfamily
\begin{tabbing}
  \hspace*{.25in}\=\hspace{2ex}\=\hspace{2ex}\=
  \hspace{2ex}\kill
  \> $k := 1$\
  \> $l_k := 0$; $r_k := 1$\
  \> loop\
  \> \> $m_k := (l_k + r_k)/2$\
  \> \> if $w < m_k$ then\
  \> \> \> $b_k := 0$; $r_k := m_k$\
  \> \> else if $w > m_k$ then\
  \> \> \> $b_k := 1$; $l_k := m_k$\
  \> \> end if\
  \> \> $k := k + 1$\
  \> end loop
\end{tabbing}
}% end \ttfamily
```

which typesets as

┌

```

k := 1
lk := 0; rk := 1
loop
  mk := (lk + rk)/2
  if w < mk then
    bk := 0; rk := mk
  else if w > mk then
    bk := 1; lk := mk
  end if
  k := k + 1
end loop
```

└

Some simple rules:



Rule:

- There is no `\\` command on a line containing the `\kill` command.
 - You may set the tabs in a `\kill` line with `\hspace` commands.
 - The `\>` command moves to the next tab stop, even if the text you have already typed extends past that stop, which can result in overprinting.
 - The `tabbing` environment has to be typeset with typewriter style font—note the `\ttfamily` command.
-

To illustrate the third rule, type

```
\begin{tabbing}
  This is short.\=\
  This is much longer, \> and jumps back.
\end{tabbing}
```

which typesets as

This is short.
This is much longer and jumps back.

If you do not follow the fourth rule, be careful with your tabbing. You do not really have to use typewriter style font—just beware of the pitfalls.

There are a number of packages that help type programming code, please consult Section 4.2 of LC3.

4.8 *Miscellaneous displayed text environments*

There are four more displayed text environments, of limited use in math:

quote, quotation, verse, verbatim.

We also discuss an inline version of the `verbatim` environment, the `\verb` command.

Quotes

The `quote` environment is used for short (one paragraph) quotations:

┌ It's not that I'm afraid to die. I just don't want to be there when
it happens. *Woody Allen*
└ Literature is news that STAYS news. *Ezra Pound*

which is typed as:

```
\begin{quote}
  It's not that I'm afraid to die. I just don't
  want to be there when it happens.
  \emph{Woody Allen}

  Literature is news that STAYS news.
  \emph{Ezra Pound}
\end{quote}
```

Note that multiple quotes are separated by blank lines.

Quotations

In the quotation environment, blank lines mark new paragraphs:

```
┌
KATH: Can he be present at the birth of his child?
    ED: It's all any reasonable child can expect if the dad is present
    at the conception.
```

Joe Orton

```
└
```

is typed as

```
\begin{quotation}
  KATH: Can he be present at the birth of his child?

  ED: It's all any reasonable child can expect
  if the dad is present at the conception.
  \begin{flushright}
    \emph{Joe Orton}
  \end{flushright}
\end{quotation}
```

Verses

A verse environment,

```
┌
I think that I shall never see
A poem lovely as a tree.

Poems are made by fools like me,
But only God can make a tree.
```

Joyce Kilmer

```
└
```


is typed as

```
\begin{verse}
  I think that I shall never see\\
  A poem lovely as a tree.

  Poems are made by fools like me,\\
  But only God can make a tree.

  \begin{flushright}
    \emph{Joyce Kilmer}
  \end{flushright}
\end{verse}
```

Lines are separated by `\\` and stanzas by blank lines. Long lines are typeset with hanging indent.

Verbatim typesetting

Finally, there is the verbatim text environment. You may need it if you write *about* L^AT_EX or some other computer programming language or if you have to include portions of a source file or user input in your typeset work. Most of the displayed source in this book was written in a verbatim environment. For instance, you may have to write to a journal about an article you are proofreading:

Formula (2) in Section 3 should be typed as follows:

```
\begin{equation}
D = \{ x_0 \mid x_0 \rightarrow a_1 \} \tag{2}
\end{equation}
```

Please make the necessary corrections.

The problem is that if you just type

Formula (2) in Section 3 should be typed as follows:

```
\begin{equation}
  D = \{ x_0 \mid x_0 \rightarrow a_1 \} \tag{2}
\end{equation}
```

Please make the necessary corrections.

it typesets as

┌

Formula (2) in Section 3 should be typed as follows:

$$(2) \qquad D = \{x_0 \mid x_0 \Rightarrow a_1\}$$

└

Please make the necessary corrections.

To get the proper typeset form, type it as follows:

Formula (2) in Section 3 should be typed as follows:

```
\begin{verbatim}
\begin{equation}
D = \{ x_0 \mid x_0 \Rightarrow a_1 \} \tag{2}
\end{equation}
\end{verbatim}
```

Please make the necessary corrections.



Rule: `verbatim` text environments

A `verbatim` environment cannot be placed within:

- another `verbatim` environment;
- the argument of a command;
- the closing line, `\end{verbatim}`, must be on a line by itself.

A violation of the first rule results in unmatched environment delimiters. You get an error message such as

```
! \begin{document} ended by \end{verbatim}.
```

A violation of the second rule gives an error message such as

```
! Argument of \@xverbatim has an extra }.
```



Rule: There are two traps to avoid when using the `verbatim` environment.

1. If the `\end{verbatim}` line starts with spaces, a blank line is added to the typeset version.
 2. Any characters following `\end{verbatim}` on the same line are dropped and you get a `LaTeX` warning.
-

To illustrate the first trap, type the last two lines of the previous example as follows:

```
\end{verbatim}
```

Please make the necessary corrections.

Then you find an unintended blank line before the last line.

The second trap can be seen if you type the last line of the above example as

```
\end{verbatim} Please make the necessary corrections.
```

When typeset, Please make the necessary corrections. does not appear, and you receive a warning

```
LaTeX Warning: Characters dropped after
'\end{verbatim}' on input line 17.
```

The `verbatim` package provides several improved versions of the `verbatim` environment. To use this package, include the command

```
\usepackage{verbatim}
```

in the preamble.

The `verbatim` environment has some interesting variants and a number of them are discussed in Section 4.2 of LC3. For instance, the `alltt` package, which is part of the standard L^AT_EX distribution is used to type the command syntax in this book. See the full syntax of `\newtheorem` on page 108.

Verbatim typesetting inline

The `verbatim` environment also has an inline version called `\verb`. Here is an example:

Some European e-mail addresses contain %;
recall that you have to type `\verb+ %+` to get %.
which prints

```
┌ Some European e-mail addresses contain %; recall that you have to type
└ \% to get %.
```

The character following the `\verb` command is a delimiter. In this example, I have used `+`. The argument starts with the character following the delimiter, and it is terminated by the next occurrence of the delimiter. In this example, the argument is `\%`.

Choose the delimiter character carefully. For instance, if you want to typeset

```

$$\sin(\pi/2 + \alpha)$$

```

verbatim, and you type

```
\verb+$\sin(\pi/2 + \alpha)$+
```

then you get the message

```
! Missing $ inserted.
<inserted text>
      $
1.5 \verb+$\sin(\pi/2 + \alpha
      )$+
```

Indeed, the argument of `\verb` is `$$\sin(\pi/2` because the second `+` terminates the `\verb` command. Then \LaTeX tries to typeset `\alpha)$+`, but cannot because it is not in math mode. Use another character, such as `!` or `|`, in place of `+`:

```
\verb!$\sin(\pi/2 + \alpha)$!
```



Rule: `\verb` command

- The entire `\verb` command must be on a single line of your source file.
- There can be no space between the `\verb` command and the delimiter.
- The `\verb` command cannot appear in the argument of another command.
- The `\verb` command cannot be used within an aligned math environment.
- Do not use `*` as a delimiter.

If you violate the first rule, as in

```
\verb!$\sin(\pi/2 +
\alpha)$!
```

you get the message

```
! LaTeX Error: \verb command ended by end of line.
```

```
1.6 \verb!$\sin(\pi/2 +
```

The `\verb` command has a `*-ed` version which prints spaces as `_` symbols. For example, `\today_`the is typed as `\verb*+\today_`the+.

The `\verb` command can perform the function of the `verbatim` environment. The last message, which was displayed in a `verbatim` environment, may be typed as follows:

```
you get the message\\[8pt]
\verb|! LaTeX Error: \verb command ended by end of line.|\\
\verb| |\\
\verb|1.6 \verb!$\sin(\pi/2 +|\\[8pt]
```



Rule: Simulating `verbatim` with `verb`

1. End the line before the `verbatim` environment with `\\[8pt]`.
2. Each line `xxx` of the `verbatim` environment is placed in the construct:

`\verb|xxx|`

If `|` occurs in `xxx`, then choose a different delimiter.

3. The last line `yyy` of the `verbatim` environment is placed in the construct:

`\verb|yyy|\\[8pt]`

If `|` occurs in `yyy`, then choose a different delimiter.

PART III

Fonts for text and math



Font basics

5.1 Shape, size, serif

Although a document class and its options determine how L^AT_EX typesets characters, there are occasions when you want control over the shape or size of the font used.

You do not have to be a typesetting expert to recognize the following basic font attributes:

Shape Normal text is typeset:

<i>upright</i> (or <i>roman</i>)	as this text
<i>slanted</i>	as this text
<i>italic</i>	as this text
<i>small caps</i>	AS THIS TEXT

Monospaced and proportional Typewriters use *monospaced* fonts, that is, fonts all of whose characters are of the same width. Most text editors display text using a monospaced font. L^AT_EX calls monospaced fonts *typewriter style*. In this book, such a font is used to represent user input and L^AT_EX's response, such as "typewriter style text". However, normal text is typeset in a *proportional* font, such as "proportional text with ii and mm," in which ii is narrow and mm is wide:

Monospaced and proportional

mmmmmm	}	monospaced
iiiiii		
 mmmmmm	}	proportional
iiiiii		

Serifs A *serif* is a small horizontal (sometimes vertical) stroke used to finish off a vertical stroke of a letter, as on the top and bottom of the letter M (large: M). L^AT_EX’s standard serif font is Computer Modern roman, such as “serif text”. Fonts without serifs are called *sans serif*, such as “sans serif text”. Sans serif fonts are often used for titles or for special emphasis.

Series: weight and width The *series* is the combination of weight and width. A font’s *weight* is the thickness of the strokes and the *width* is how wide the characters are.

Light, *medium* (or *normal*), and *bold* often describe weight.

Narrow (or *condensed*), *medium* (or *normal*), and *extended* often describe width.

The Computer Modern family includes **bold fonts**. Traditionally, when the user asks for bold CM fonts, L^AT_EX actually provides *bold extended* (a somewhat wider version).

Size Most L^AT_EX articles are typeset with 10 point text unless otherwise instructed. Larger sizes are used for titles, section titles, and so on. Abstracts and footnotes are often set in 8-point type.

Font family The collections of all sizes of a font is called a *font family*.

5.2 Document font families

In a document class, the style designer designates three document font families:

1. *Roman* (upright and serified) document font family,
2. *Sans serif* document font family,
3. *Typewriter style* document font family,

and picks one of these (for articles, as a rule, the roman document font family) as the *document font family* or *normal family*. In all the examples in this book, the document font family is the roman document font family except for presentations that use sans serif (see Section 1.11 and Chapter 13). When you use Computer Modern fonts in L^AT_EX, which is the default, the three document font families are Computer Modern roman, Computer Modern sans serif, and Computer Modern typewriter. The document font family is Computer Modern roman.

In this book, the roman document font family is Times, the sans serif document font family is Helvetica, and the typewriter style document font family is Computer Modern typewriter. The document font family is the roman document font family Times.

The document font family (normal family) is the default font. You can always switch back to it with

```
\textnormal{...}
```

or

```
{\normalfont ...}
```

Table 5.1 shows these two commands and three additional pairs of commands to help you switch among the three basic document font families. It also shows the command pairs for the basic font shapes.

Command with argument	Command declaration	Switches to the font family
<code>\textnormal{...}</code>	<code>{\normalfont ...}</code>	document
<code>\emph{...}</code>	<code>{\em ...}</code>	<i>emphasis</i>
<code>\textrm{...}</code>	<code>{\rmfamily ...}</code>	roman
<code>\textsf{...}</code>	<code>{\sffamily ...}</code>	sans serif
<code>\texttt{...}</code>	<code>{\ttfamily ...}</code>	typewriter style
<code>\textup{...}</code>	<code>{\upshape ...}</code>	upright shape
<code>\textit{...}</code>	<code>{\itshape ...}</code>	<i>italic shape</i>
<code>\textsl{...}</code>	<code>{\slshape ...}</code>	<i>slanted shape</i>
<code>\textsc{...}</code>	<code>{\scshape ...}</code>	SMALL CAPITALS
<code>\textbf{...}</code>	<code>{\bfseries ...}</code>	bold
<code>\textmd{...}</code>	<code>{\mdseries ...}</code>	normal weight and width

Table 5.1: Font family switching commands.

The font-changing commands of Table 5.1 come in two forms:

- A command with an argument, such as `\textrm{...}`, changes its argument. These are short commands, that is, they cannot contain a blank line or a `\par` command.
- A command declaration, such as `\rmfamily`, carries out the font change following the command and within its scope (see Section 3.3.2).

You should always use commands with arguments for small changes within a paragraph. They have two advantages:

- You are less likely to forget to change back to the normal font.
- You do not have to worry about italic corrections (see Section 5.3.1).

Note that *MakeIndex* requires that you use commands with arguments to change the font in which page numbers are typeset (see Section 18.1).

For font changes involving more than one paragraph, use command declarations. These commands are preferred if you want to create custom commands and environments (see Chapter 15).

5.3 Shape commands

There are five pairs of commands to change the font shape:

- `\textup{...}` or `{\upshape ...}` switch to the upright shape.
- `\textit{...}` or `{\itshape ...}` switch to the *italic shape*.
- `\textsl{...}` or `{\slshape ...}` switch to the *slanted shape*.
- `\textsc{...}` or `{\scshape ...}` switch to SMALL CAPITALS.
- `\emph{...}` or `{\em ...}` switch to *emphasis*.

The document class specifies how emphasis is typeset. As a rule, it is italic or slanted unless the surrounding text is italic or slanted, in which case it is upright. For instance,

`\emph{Rubin space}`

in the statement of a theorem is typeset as

┌
the space satisfies all three conditions, a so-called Rubin space that ...
└

The emphasis changed the style of Rubin space from italic to upright.



Tip Be careful not to interchange the command pairs. For instance, if by mistake you type `{\textit serif}`, the result is serif. Only the *s* is italicized since `\textit` takes *s* as its argument.



Rule: Abbreviations and acronyms

For abbreviations and acronyms use small caps, except for two-letter geographical acronyms.

So Submitted to TUG should be typed as

Submitted to `\textsc{tug}`

Note that only the lowercase characters in the argument of the `\textsc` command are printed as small caps.

5.3.1 *Italic corrections*

The phrase

when using a *serif* font

may be typed as follows:

when using a `{\itshape serif\}` font

The `\/` command before the closing brace is called an *italic correction*. Notice that `{\itshape M}M` typesets as *MM*, where the *M* is leaning into the *M*. Type `{\itshape M\/}M` to get the correct spacing *MM*. Compare the typeset phrase from the previous example with and without an italic correction:

when using a *serif* font
when using a *serif* font

The latter is not as pleasing to the eye.



Rule: 1. Italic correction

If the emphasized text is followed by a period or comma, you should not type the italic correction.

For example,

Do not forget. My party is on Monday.

should be typed as

`{\itshape Do not forget.} My party is on Monday.`



Rule: 2. Italic correction

The shape commands with arguments do not require italic correction. The corrections are provided automatically where needed.

Thus you can type the phrase when using a *serif* font the easy way:

when using a `\textit{serif}` font

Whenever possible, let L^AT_EX take care of the italic correction. However, if L^AT_EX is adding an italic correction where you feel it is not needed, you can override the correction with the `\nocorr` command. L^AT_EX does not add an italic correction before a period or a comma. These two punctuation marks are stored in the `\nocorrlist` command. By redefining this command, you can modify L^AT_EX's behavior.



Rule: 3. Italic correction

The italic correction is required with the commands `\itshape`, `\slshape`, and `\em`.

5.3.2 Series

These attributes play a very limited role with the Computer Modern fonts. There is only one important pair of commands,

```
\textbf{...}    {\bfseries ...}
```

to change the font to bold (actually, bold extended). The commands

```
\textmd{...}    {\mdseries ...}
```

which set both the weight and width to medium (normal) are seldom needed.

5.4 Size changes

Standard L^AT_EX documents are typeset in 10 point type. The 11 point and 12 point type are often used for better readability; some journals require 12 point type—if this is the case, use the 12pt document class option (see Sections 11.5 and 19.1.3). The 8pt and 9pt document class options are rarely used. The sizes of titles, subscripts, and superscripts are automatically set by the document class, in accordance with the font size option.

If you must change the font size for some text—it is seldom necessary to do so in an article—the following command declarations are provided (see Table 5.2):

```
\Tiny \tiny \SMALL \Small \small
      \normalsize
\large \Large \LARGE \huge \Huge
```

The command `\SMALL` is also called `\scriptsize` and the command `\Small` is also called `\footnotesize`. The font size commands are listed in order of increasing—to be more precise, nondecreasing—size.

Command	Sample text
<code>\Tiny</code>	sample text
<code>\tiny</code>	sample text
<code>\SMALL</code> or <code>\scriptsize</code>	sample text
<code>\Small</code> or <code>\footnotesize</code>	sample text
<code>\small</code>	sample text
<code>\normalsize</code>	sample text
<code>\large</code>	sample text
<code>\Large</code>	sample text
<code>\LARGE</code>	sample text
<code>\huge</code>	sample text
<code>\Huge</code>	sample text

Table 5.2: Font size commands.

Two commands allow the user to increase or decrease font size: `\larger` moves up one size, `\smaller` moves down one. Both commands may take an optional argument. For example, `\larger[2]` moves up 2 sizes.

5.5 Orthogonality

You are now familiar with the commands that change the font family, shape, series, and size. Each of these commands affects one and only one font attribute. For example, if you change the series, then the font family, shape, and size do not change. These commands act independently. In \LaTeX terminology, the commands are *orthogonal*. From the user's point of view this behavior has an important consequence: *The order in which these commands are given does not matter*. Thus

```
\Large \itshape \bfseries
```

has the same effect as

```
\bfseries \itshape \Large
```

Note that \LaTeX 2.09's two-letter commands (see Section 5.6) are not orthogonal.

Orthogonality also means that you can combine these font attributes in any way you like. For instance, the commands

```
\sffamily \slshape \bfseries \Large
```

instruct \LaTeX to change the font family to sans serif, the shape to slanted, the series to bold, and the size to `\Large`. If the corresponding font is not available, \LaTeX uses a font that is available, and issues a warning.

The font substitution algorithm may not provide the font you really want, so it is your responsibility to make sure that the necessary fonts are available. We discuss this topic further in Section 19.4.

5.6 *Obsolete two-letter commands*

Early users of L^AT_EX are accustomed to using the two-letter commands `\bf`, `\it`, `\rm`, `\sc`, `\sf`, `\sl`, and `\tt`. These commands are no longer part of L^AT_EX. They are, however, still defined in most document classes. The two-letter commands:

1. switch to the document font family;
2. change to the requested shape.

There are a number of reasons not to use them. The two-letter commands:

- are not part of L^AT_EX;
- require manual italic corrections;
- are not orthogonal (see Section 5.5).

`\slshape \bfseries` is the same as `\bfseries \slshape` (slanted bold), but `\sl\bf` is not the same as `\bf\sl`. Indeed, `{\sl\bf sample}` gives **sample** and `{\bf\sl sample}` produces *sample*.



Font encoding

6.1 OT1, T1, ...

A font has *glyphs* (letters, digits, symbols) available for printing. \TeX refers to a glyph of a font by addressing it with an 8-bit number. Such a mapping is a *font encoding*. The fonts originally distributed with \TeX had only 128 glyphs per font. The name of this encoding is OT1 (Old Text encoding); see Figure 6.1.

T1 is an alternative encoding that enables \TeX to typeset correctly (with proper hyphenation and kerning) in more than 30 languages based on the Latin alphabet; see Section 6.4.

T1 was superseded by UTF-8 (Unicode Transformation Format), which represents all characters in the Unicode standard (as of 2021 there are 143,859 characters). It became available in the late 1990s; however, it wasn't until the release of \TeX Live 2018 that LaTeX natively supported UTF-8. This encoding is now widely used and has become the default character encoding in all operating systems and also on the Web.

I would still suggest to start all your \LaTeX documents with

```
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
```

Character	OT1 code
A	65
B	66
C	67
...	...
Z	90
a	97
b	98
c	99
...	...
z	122

Figure 6.1: OT1 encoding

in case your colleague uses an old version of \LaTeX . A friend purchased Adobe Illustrator around 2010, which has not been available for purchase since 2013. To keep it working, he still uses Windows 7!

6.2 Low-level commands

Font attributes are changed using the low-level commands

`\fontencoding`, `\fontfamily`, `\fontseries`, `\fontshape`, `\fontsize`,

and so on. The `\selectfont` command changes the font. The low-level commands have been developed for document class and package writers. However, some familiarity with these is useful.

There is one use of low-level commands you should keep in mind. When you choose a font size for your document or for some part thereof, you also determine the `\baselineskip`, the distance from the baseline of one line to the baseline of the next. Typically, a 10-point font size uses a 12 point `\baselineskip`. Occasionally, you may want to change the font size along with the `\baselineskip`. For instance, the command

```
\fontsize{9pt}{11pt}\selectfont
```

changes the font size to 9 point and the `\baselineskip` to 11 point. To make this change for a single paragraph, you can type

```
{%special paragraph
\fontsize{9pt}{11pt}\selectfont
text
```

```
%end special paragraph
```


6.3 PostScript fonts

The Computer Modern fonts were originally “hardwired” into \LaTeX . Many users liked \LaTeX but not the Computer Modern font. With the spread of personal computers and PostScript laser printers, PostScript fonts had to be integrated into \LaTeX . I will describe how easy it is to use standard PostScript fonts, such as Times and how to replace the CM and AMS fonts in a \LaTeX document with the Lucida Bright fonts.

You can obtain PostScript fonts from a wide variety of sources. There are many free PostScript fonts on CTAN. LC3 dedicates more than a 100 pages discussing them. Table 6.1 is a short list of the more prominent commercial vendors.

Foundry	URL
Adobe	www.adobe.com/type/
Agfa/Monotype	www.agfamonotype.com/
Berthold	www.myfonts.com/
Bitstream	www.bitstream.com/
Emigre	www.emigre.com/
Hoefer	www.typography.com/
ITC	www.itcfonts.com/
Linotype	www.linotype.com/
Monotype	www.fonts.com/
Scriptorium	www.fontcraft.com/
Vintage	www.vintagetype.com/

Table 6.1: Some type foundries on the Internet.

For Berthold, choose H. Berthold Typefoundry in myfonts.com.

See also the Web page at <http://www.microsoft.com/typography/> for a lot of useful information and links.

The Times font and MathTime

In this section, we step through the process of incorporating the Adobe Times font into a \LaTeX document to replace the Computer Modern text fonts, and, optionally, of using the *MathTime Pro 2* math fonts to replace the Computer Modern math fonts.

A document class specifies three standard font families (see Section 5.2):

- a roman (or serif) font family,
- a sans serif font family,
- a typewriter style font family.

The `times` package in the PSNFS distribution makes Times the roman font family, Helvetica the sans serif font family, and Courier the typewriter style font family.

Setting up Times

First, install the Adobe Times, Helvetica, and Courier PostScript fonts and their \TeX font metric files.

Now typeset the `psfonts.ins` file—in the PSNFSS distribution. This produces `sty` files for the standard PostScript fonts. The Times style file is called `times.sty`. If you do not already have it, copy it into a folder \LaTeX can access.

Using Times

In the preamble of your document, type

```
\usepackage{times}
```

after the `\documentclass` line. Then Times becomes the roman, Helvetica the sans serif, and Courier the typewriter style document font family. That is all there is to it.

Using the `times` package changes the document font family throughout your document. To switch to Times only occasionally, type

```
{\fontfamily{ptm}\selectfont phrase}
```

The text preceding and following this construct is not affected. For example,

```
{\fontfamily{ptm}\selectfont
This text is typeset in the Times font.}
```

typesets as

```
┌
└ This text is typeset in the Times font.
```

Similarly,

```
\fontfamily{ptm}\selectfont
This text is typeset in the Times font.
\normalfont
```

also typesets the same phrase in Times. Recall that the `\normalfont` command restores the document font family (see Section 5.2).

Lucida Bright fonts

Another alternative to Computer Modern fonts is *Lucida Bright* for both text and math fonts. You can purchase the Lucida Bright fonts from TUG. LC3 is typeset in Lucida.

Copy the files

```
lucidabr.ins, lucidabr.dtx,
lucidabr.fdd,lucidabr.yy
```

into your T_EX input folder. Typeset `lucidabr.yy`, producing the `lucidabr.sty` file and a large number of `fd` files.

Now add the lines

```
\usepackage[LY1]{fontenc}    %specify font encoding
\usepackage[LY1]{lucidabr}    %switch text and math fonts
```

in the preamble of your document. The `lucidabr` package has many options. See its documentation—typeset `lucidabr.dtx` to get it.

6.4 *L^AT_EX localized*

If you write articles is not in American English and/or your keyboard is not the standard American keyboard, you may find it annoying, and sometimes difficult, to use standard L^AT_EX. The annoyance may start with finding out how to type `~` for a nonbreakable space, to L^AT_EX's inability to properly hyphenate `Gr\{"a}tzer`, and L^AT_EX's inability to use a different alphabet.

To localize L^AT_EX, that is, to adapt L^AT_EX for use with languages other than American English and keyboards other than standard American keyboards, the major players are the `babel`, `fontenc`, `inputenc` packages, along with font-encoding schemes.

The `babel` package is described in detail in Johannes Braams [3] and in Chapter 13 of LC3

If you are interested in using a localized L^AT_EX, you should turn to the T_EX user group for that linguistic group to find out what is available. You should also consult the `babel` user guide.

At a minimum, a supported language has translated redefinable names (see Table 15.1), and a localized variant of the `\today` command. Two very advanced language adaptations are German and French.

We first illustrate the use of the `babel` package with the German language, which gives you a rich set of features, including typing:

- "a for `\{"a}`,
- "s for sharp s (eszett),
- "ck for a ck that becomes k-k when hyphenated.

Type the following test file: (`german.tex` in the `samples` folder):

```
\documentclass{amsart}
\usepackage[german]{babel}

\begin{document}
\section{H"ullenoperatoren}
Es sei  $\mathbb{P}$  eine teilweise geordnete Menge. Wir sagen,
```

```
dass in  $P$  ein  $\lambda$ 
erkl"art ist, wenn sich jedem  $a \in P$  ein eindeutig
bestimmtes  $\lambda(a) \in P$  zuordnen l"a"st, so dass
die folgenden Bedingungen erf"ullt sind.
\end{document}
```

And here it is typeset:

1 Hüllenoperatoren

Es sei P eine teilweise geordnete Menge. Wir sagen, dass in P ein *Hüllenoperator* λ erklärt ist, wenn sich jedem $a \in P$ ein eindeutig bestimmtes $\lambda(a) \in P$ zuordnen läßt, so dass die folgenden Bedingungen erfüllt sind.

The second example for babel uses the following options for the packages:

```
\usepackage[T2A]{fontenc}
\usepackage[koi8-u]{inputenc}
\usepackage[ukrainian]{babel}
```

koi8-u is the appropriate encoding for Ukrainian.

And here is the typeset Ukrainian file:

Поняття теорії ігор

Віктор Анякін

31 липня 2022 р.

Логічною основою теорії ігор є формалізація трьох понять, які входять в її визначення і є фундаментальними для всієї теорії:

- Конфлікт,
- Прийняття рішення в конфлікті,
- Оптимальність прийнятого рішення.

Ці поняття розглядаються в теорії ігор у найширшому сенсі. Їх формалізації відповідають змістовним уявленням про відповідні об'єкти.

PART IV

Math into L^AT_EX



Typing math

L^AT_EX was designed for typesetting math. I address this topic in detail.

A math formula can be typeset *inline*, as part of the current paragraph, or *displayed*, on a separate line or on lines with vertical space before and after the formula.

In this and the next chapter we discuss formulas that are set inline or displayed on a *single line*. In Chapter 9 we address *multiline* math formulas.

We start with a discussion of L^AT_EX's basic math commands (Section 7.1), spacing rules in math (Section 7.2), and continue with the `equation` environment (Section 7.3). The basic constructs of a formula—arithmetic (including subscripts and superscripts), binomial coefficients, ellipses, integrals, roots, and text—are discussed in detail in Section 7.4. From the basic constructs of that section, you can build very complicated formulas, one step at a time. The process is illustrated in Section 7.9.

Delimiters, operators, and math accents are dealt with in Sections 7.5–7.7. In Section 7.8, we discuss three types of stretchable horizontal lines that can be used above or below a formula: braces, bars, and arrows. There are also stretchable arrow math symbols.

Section 7.10 is our *Formula Gallery*, in which you find a large number of illustrations, some straightforward, some more imaginative, of the math constructs introduced in the preceding sections.

7.1 Math environments

A formula in a \LaTeX document can be typeset *inline*, like the congruence $a \equiv b \ (\theta)$ or the integral $\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$, or *displayed*, as in

$$a \equiv b \ (\theta)$$

or

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

Notice how changing these two formulas from inline to displayed affects their appearance.

Inline and displayed math formulas are typeset using the *math environments* `math` and `displaymath`, respectively. Because math formulas occur so frequently, \LaTeX has abbreviations: the special braces `\(` and `\)` or `$` are used for the `math` environment, and `\[` and `\]` for the `displaymath` environment. So our inline example may be typed as

```
$a \equiv b \ \text{\pod{\theta}}$
```

or

```
\( a \equiv b \ \text{\pod{\theta}} \)
```

or

```
\begin{math}
  a \equiv b \ \text{\pod{\theta}}
\end{math}
```

The displayed example can be typed as

```
\[
  \int_{-\infty}^{\infty} e^{-x^2} \, dx = \sqrt{\pi}
\]
```

or

```
\begin{displaymath}
  \int_{-\infty}^{\infty} e^{-x^2} \, dx = \sqrt{\pi}
\end{displaymath}
```

Using `$` as a delimiter for a `math` environment is a bit of an anomaly, since the same character is used as both an opening and closing delimiter. This dual purpose use makes it more difficult for \LaTeX to diagnose an incorrect use of `$`. For instance,

Let `$a` be a real number, and let `f` be a function.

would be interpreted by \LaTeX as follows:

- `Let` is ordinary text
- `$a` be a real number, and let `$` is math
- `f` is interpreted as ordinary text
- `$` be a function. is thought to be a math environment (opened by `$`) that should be closed by the next `$` in the paragraph

Because the paragraph ends with no more dollar signs appearing, you get the message

```
!! Missing $ inserted.
<inserted text>
      $
```

1.29

and giving you the line number of the end of the paragraph. This message tells you that \LaTeX would place a `$` at the end of the paragraph when it proceeds with the typesetting. Press Return; \LaTeX produces the following:

```
[
Let abearealnumber, andletfbeafunction.
]
```

The text that ended up in a math environment is run together because math environments ignore spaces (see Section 7.2).

If you use `\(` and `\)` as special braces for the math environment, \LaTeX handles the same mistake more elegantly:

```
Let \(( a be a real number, and let \(( f \)) be a function.
gives the message
```

```
! LaTeX Error: Bad math environment delimiter.
```

```
1.25 Let \(( a be a real number, and let \((
      f \)) be a function.
```

\LaTeX realizes that the first `\(` opens a math environment, so the second `\(` must be in error. In this case, the line number in the message is correct.

Throughout this book, like nearly everyone else, I use `$` to delimit inline math.

\TeX uses `$$` to open and close a displayed math environment. In \LaTeX , this may occasionally cause problems. Don't do it! Try the `fleqn` document class option of `amsart` (see Section 11.5) as an example of what can go wrong.



Rule: Math environments

No blank lines are permitted in a `math` or `displaymath` environment.

If you violate this rule, \LaTeX generates a message,

```
! Missing $ inserted.
```

```
<inserted text>
```

```
      $
```

```
...
```

```
1.7
```

where the line number points inside the environment.

Multiline math environments, such as the examples in Sections 1.7.3 and 1.7.4, are discussed in Chapter 9.

7.2 Spacing rules

In text, the most important spacing rule is that any number of spaces in the source file equals one space in the typeset document. The spacing rule for math mode is even more straightforward.



Rule: Spacing in math

\LaTeX ignores spaces in math.

In other words, all spacing in math mode is provided by \LaTeX . For instance,

```
$a+b=c$
```

and

```
$a + b = c$
```

are both typeset as $a + b = c$.

There are two exceptions to this rule:

1. A space indicating the end of a command name is recognized. For instance, in

```
$a \quad b$
```

\LaTeX does not ignore the space between `\quad` and `b`.

2. If you switch back to text mode inside a math formula with a `\text` command (see Section 7.4.6), then the text spacing rules apply in the argument of such a command.

As you see, \LaTeX provides controls for spaces in typeset math. The spaces you type in math do not affect the typeset document. But keep this tip in mind.



Tip Format your source file so that it is easy to read.

When typing a source file, the following is good practice:

- Place `\[` and `\]` on lines by themselves.
- Leave spaces before and after binary operations and binary relations, including the equal sign.
- Indent—by three spaces, for example—the contents of environments so they stand out.
- Keep a formula on a single line of the source file, if you can.

Develop your own style of typing math, and stick with it.



Tip The spacing after a comma is different in math and text. Do not leave a trailing comma in inline math.

So do not type

If $a = b, \$$ then

but move the comma out.

7.3 Equations

An *equation* is a numbered formula displayed on a single typeset line.

Equations are typed in an `equation` environment. The `equation` environment and `displaymath` environment are exactly the same except that the `equation` environment assigns a number to the displayed formula

$$(1) \quad \int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

This example is typed as

```
\begin{equation}\label{E:int}
\int_{-\infty}^{\infty} e^{-x^2} \, dx = \sqrt{\pi}
\end{equation}
```

The `\label` command in the `equation` environment is optional. If you use a `\label` command, the number assigned to the equation can be referenced with the `\ref` command. So

```
see~(\ref{E:int})
```

typesets as see (1). Even better, use the `\eqref` command, which places the parentheses automatically:

```
see~\eqref{E:int}
```

also typesets as *see* (1). In fact, the `\eqref` command does more: It typesets the reference *upright*, even in italicized or slanted text. For more information about cross-referencing; see Section 1.7.2.

As a rule, equations are numbered consecutively throughout articles, whereas in books, numbering starts from 1 at the start of each chapter. You may also choose to have equations numbered within each section—(1.1), (1.2), ..., in Section 1; (2.1), (2.2), ..., in Section 2; and so on—by including the command

```
\numberwithin{equation}{section}
```

in the preamble of your document (see Section 10.2). “Manual control” of numbering is discussed in Section 15.5.1, group numbering in Section 9.4.4.

The `*`-ed form of the `equation` environment suppresses numbering, so it is equivalent to the `displaymath` environment (or the special braces `\[` and `\]`).



Rule: Equation environment

1. No blank lines are permitted within an `equation` or `equation*` environment.
2. No blank line before the environment.

If you typeset

```
\begin{equation}\label{E:int}
\int_{-\infty}^{\infty} e^{-x^2} \, dx = \sqrt{\pi}

\end{equation}
```

LaTeX generates the familiar, but misleading, message

```
! Missing $ inserted.
```

7.4 Basic constructs

A formula is built by combining various basic constructs. This section discusses the following constructs:

- arithmetic operations, – subscripts and superscripts,
- binomial coefficients,
- ellipses,

- integrals,
- roots,
- text,
- Hebrew and Greek letters.

Read carefully the basic constructs *important for your work*. Additional constructs are discussed in subsequent sections of this chapter.

7.4.1 Arithmetic operations

The *arithmetic operations* are typed pretty much as you would expect. To get $a + b$, $a - b$, $-a$, a/b , and ab , type

`$a + b$`, `$a - b$`, `$-a$`, `a / b`, `$a b$`

There are two other forms of multiplication and one of division: $a \cdot b$, $a \times b$, and $a \div b$. They are typed as follows:

`$a \cdot b$`, `$a \times b$`, `$a \div b$`

In displayed formulas, *fractions* are usually typed with the `\frac` command. To get

$$\frac{1 + 2x}{x + y + xy}$$

type

```
\[
  \frac{1 + 2x}{x + y + xy}
\]
```

You can use display-style fractions inline with `\dfrac`, and inline-style fractions in displayed math environments with `\tfrac`; for example, $\frac{3 + a^2}{4 + b}$ is typed as

`$\dfrac{3 + a^2}{4 + b}$`

and

$$\frac{3+a^2}{4+b}$$

is typed as

```
\[
  \tfrac{3 + a^2}{4 + b}
\]
```

The `\dfrac` command is often used in matrices whose entries would look too small with the `\frac` command. See Formula 20 in the *Formula Gallery* (Section 7.10) for an example, and Section 8.7 for other fraction variants.

Subscripts and superscripts

Subscripts are typed with `_` and *superscripts* with `^`. Remember to enclose the subscripted or superscripted expression in braces:

```
\[
  a_{1},\ a_{i_{1}},\ a^{2},\ a^{b^{c}},\ a^{i_{1}},\
    a_{i} + 1,\ a_{i + 1},\ a_{1}^{2},\ a^{2}_{1}
\]
```

typesets as

$$a_1, a_{i_1}, a^2, a^{b^c}, a^{i_1}, a_i + 1, a_{i+1}, a_1^2, a_1^2$$

For a^{b^c} , type `$a^{b^{c}}$`, not `$a^{b}{c}$`. If you type the latter, you get the message

! Double superscript.

Similarly, a_{b_c} is typed as `$a_{b_{c}}$`, not as `a_{b}_{c}`.

In many instances, the braces for the subscripts and superscripts could be omitted, but you should type them anyway.



Tip You may safely omit the braces for a subscript or superscript that is a single digit or letter, as in `a_1` and `$(a + b)^x$`, which are typeset as a_1 and $(a + b)^x$. Be careful, however, if you have to edit `a_1` to make it a_{12} , then the braces can no longer be omitted, you must type `a_{12}` to obtain a_{12} because `a_12` typesets as a_12 .

There is one symbol that is automatically superscripted in math mode, the prime, that is, `'`. To get $f'(x)$, type `$f'(x)$`. However, to get f'^2 you must type

```
$f^{\prime 2}$
```

Typing `$(f')^2$` results in f'^2 , with the 2 too high; typing it as `f'^{2}` causes a double superscript error. Sometimes you may want a symbol to appear superscripted or subscripted by itself, as in the phrase

┌ use the symbol \dagger to indicate the dualspace
└

typed as

```
use the symbol {}^{\dagger} to indicate the dualspace
```

where `{ }` is the *empty group*. The empty group can be used to separate symbols, to terminate commands, or as the base for subscripting and superscripting.

The `\sb` and `\sp` commands also typeset subscripts and superscripts, respectively, as in

`$a\sb{1} - a\sp{x + y}$`

which produces $a_1 - a^{x+y}$. These commands are seldom used.

For multiline subscripts and superscripts; see Section 7.6.4.

7.4.2 Binomial coefficients

Binomials are typeset with the `\binom` command. Here are two examples shown inline,

$\binom{a}{b+c}$ and $\binom{\frac{n^2-1}{2}}{n+1}$, and displayed:

$$\binom{a}{b+c} \text{ and } \binom{\frac{n^2-1}{2}}{n+1}$$

The latter is typed as

```
\[
  \binom{a}{b + c} \text{ and }
  \binom{\frac{n^2}{2} - 1}{n + 1}
\]
```

You can use display-style binomials inline with `\dbinom`, and inline-style binomials in displayed math environments with `\tbinom`. For example, $\binom{a}{b+c}$ is typed as `$\dbinom{a}{b + c}$` (looks awful). See Section 8.7 for other variants.

7.4.3 Ellipses

There are two types of *ellipsis* in math, the *low* or *on-the-line ellipsis*, as in

$$F(x_1, x_2, \dots, x_n)$$

and the *centered ellipsis*, as in

$$x_1 + x_2 + \cdots + x_n$$

These two formulas are typed as

```
\[
  F(x_{1}, x_{2}, \dots, x_{n})
\]
```

and

```
\[
  x_{1} + x_{2} + \dots + x_{n}
\]
```

L^AT_EX uses the symbol following a `\dots` command to decide whether to use a low or centered ellipsis. If it fails to make the right decision as in

$$\alpha(x_1 + x_2 + \dots)$$

typed as

```
\[
  \alpha(x_{1} + x_{2} + \dots)
\]
```

help L^AT_EX by giving the command `\ldots` for low and `\cdots` for centered ellipsis. So to get the last formula right, type

```
\[
  \alpha(x_{1} + x_{2} + \cdots)
\]
```

and it typesets correctly:

$$\alpha(x_1 + x_2 + \cdots)$$

As I discuss it in Section 15.3, I use the custom command `\jj` for `\vee` (which typesets as \vee). To get

$$x_1 \vee x_2 \vee \cdots \vee x_n$$

do not type

```
\[
  x_{1} \jj x_{2} \jj \dots \jj x_{n}
\]
```

which typesets as

$$x_1 \vee x_2 \vee \dots \vee x_n$$

Instead, type

```
\[
  x_{1} \jj x_{2} \jj \cdots \jj x_{n}
\]
```

There are five more variants of the `\dots` command:

- `\dotsc` for an ellipsis followed by a comma,
- `\dotsb` for an ellipsis followed by a binary operation or relation,
- `\dotsm` for an ellipsis followed by multiplication,
- `\dotsi` for an ellipsis with integrals,
- `\dotso` for an “other” ellipsis.

These commands not only force the ellipsis to be low or centered, but also adjust the spacing.

See Section 9.7.1 for an example of *vertical dots* with the `\vdots` command and *diagonal dots* with the `\ddots` command.

7.4.4 Integrals

You have already seen the formula $\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$ in both inline and displayed forms in the first section of this chapter. The lower limit is typeset as a subscript and the upper limit is typeset as a superscript. To force the limits below and above the integral symbol inline, use the `\limits` command. The `\nolimits` command does the reverse. To typeset $\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$, type

`\int\limits_{-\infty}^{\infty} e^{-x^2} \, dx = \sqrt{\pi}`

See Section 11.5 for a discussion of the `intlimits` document class option.

There are five commands to produce variants of the basic integral symbol:

`\oint` `\iint` `\iiint` `\iiiint` `\idotsint`

which typeset as

$$\oint \quad \iint \quad \iiint \quad \iiiint \quad \int \cdots \int$$

For complicated bounds, use the `\substack` command or the `subarray` environment (see Section 7.6.4).

7.4.5 Roots

The `\sqrt` command produces a square root, for instance,

`\sqrt{5}` typesets as $\sqrt{5}$
`\sqrt{a + 2b + c^2}` typesets as $\sqrt{a + 2b + c^2}$

Here is a more interesting example:

$$\sqrt{1 + \sqrt{1 + \frac{1}{2}\sqrt{1 + \frac{1}{3}\sqrt{1 + \frac{1}{4}\sqrt{1 + \cdots}}}}}$$

typed as

`\[`
`\sqrt{1 + \sqrt{1 + \frac{1}{2}\sqrt{1 + \frac{1}{3}\sqrt{1 + \frac{1}{4}\sqrt{1 + \cdots}}}}}`
`\]`

For n -th roots other than the square root, that is, $n \neq 2$, specify n with an optional argument. To get $\sqrt[3]{5}$, type `\sqrt[3]{5}`.

Root refinement

In $\sqrt[5]{g}$, typed as `\sqrt[5]{g}`, the placement of g is not very pleasing. \LaTeX provides two additional commands to allow you to adjust the position of g :

`\leftroot` moves g to the left—or to the right with a negative argument;

`\uproot` moves g up—or down with a negative argument.

You may prefer one of the following variants:

$\sqrt[5]{g}$ typed as `\sqrt[\leftroot{2}\uproot{2}]{g}`

$\sqrt[5]{g}$ typed as `\sqrt[\uproot{2}]{g}`

Experiment with `\leftroot` and `\uproot` to find the best spacing.

Note that \LaTeX is very finicky with this optional argument. Typing a space after `[`, as in `\sqrt[\uproot{2}]{g}`, gives the message

! Package amsmath Error: Invalid use of \uproot.

There may also be problems with vertical spacing under the root symbol (see Section 8.5).

7.4.6 Text in math

In the `amstex` class (in general, with the `amsmath` package), \LaTeX allows you to include text in formulas with the `\text` command. The formula

$$A = \{x \mid x \in X_i, \text{ for some } i \in I\}$$

is typed as

```
\[
  A = \{ x \mid x \in X_{\text{\textit{i}}}, \text{\textit{for some }}
        \text{\textit{i}} \in I \}
\]
```

Note that you have to leave space before `for` and after `some` inside the argument of `\text`. The argument of the `\text` command is always typeset in a single line.

Sometimes it is more convenient to go into math mode within the argument of a `\text` command rather than end the `\text` and start another, as in

$$A = \{x \mid \text{for } x \text{ large}\}$$

which may be typed as

```
\[
  A = \{ x \mid \text{\textit{for }} \$x\$ \text{\textit{large}} \}
\]
```

The `\text` command correctly sizes its argument to match the context. The formula

$$a_{\text{left}} + 2 = a_{\text{right}}$$

is typed as

```
\[
  a_{\text{left}} + 2 = a_{\text{right}}
\]
```

Note that `\text` typesets its argument *in the size and shape* of the surrounding text. If you want the text in a formula to be typeset in the document font family (see Section 5.2) independent of the surrounding text, use

```
\textnormal{ ... }
```

or

```
{\normalfont ...}
```

For instance, if you have a constant a_{right} , then in a theorem:

Theorem 1. *The constant a_{right} is recursive in a .*

The subscript is wrong. To get it right, type the constant as

```
$a_{\normalfont\text{right}}$
```

Now the theorem typesets as

Theorem 1. *The constant a_{right} is recursive in a .*

Any of the text font commands with arguments (see Section 5.3) can also be used in math formulas. For instance, `\textbf` uses the size and shape of the surrounding text to typeset its argument in bold (extended).



Rule: If a is subscripted l for left, as in a_l , type it as `a_{l}`. Unfortunately, many papers use a_l , typed as `a_l`.

If in a_l , the character l is text, type it as `a_{l}`.

If in a_l , the character l is a variable, type it as `a_l`.

7.4.7 *Hebrew and Greek letters*

Math uses only four Hebrew letters: א, ב, ג, ד, typed as

`\aleph`, `\beth`, `\daleth`, `\gimel`

The 26 Greek letters come in lower case and some also in upper case. There is no upper case α , because it is the same as the latin letter A. Seven lower case Greek letter also come in a variant. For instance, the variant of ϕ is φ . Tables 7.1 and 7.2 list them all; see also Appendix A.1.

Type	Typeset	Type	Typeset	Type	Typeset
<code>\alpha</code>	α	<code>\iota</code>	ι	<code>\sigma</code>	σ
<code>\beta</code>	β	<code>\kappa</code>	κ	<code>\tau</code>	τ
<code>\gamma</code>	γ	<code>\lambda</code>	λ	<code>\upsilon</code>	υ
<code>\delta</code>	δ	<code>\mu</code>	μ	<code>\phi</code>	ϕ
<code>\epsilon</code>	ϵ	<code>\nu</code>	ν	<code>\chi</code>	χ
<code>\zeta</code>	ζ	<code>\xi</code>	ξ	<code>\psi</code>	ψ
<code>\eta</code>	η	<code>\pi</code>	π	<code>\omega</code>	ω
<code>\theta</code>	θ	<code>\rho</code>	ρ		
<code>\varepsilon</code>	ε	<code>\varpi</code>	ϖ	<code>\varsigma</code>	ς
<code>\vartheta</code>	ϑ	<code>\varrho</code>	ϱ	<code>\varphi</code>	φ
	<code>\digamma</code>	F	<code>\varkappa</code>	\varkappa	

Table 7.1: Lowercase Greek letters

Type	Typeset	Type	Typeset	Type	Typeset
<code>\Gamma</code>	Γ	<code>\Xi</code>	Ξ	<code>\Phi</code>	Φ
<code>\Delta</code>	Δ	<code>\Pi</code>	Π	<code>\Psi</code>	Ψ
<code>\Theta</code>	Θ	<code>\Sigma</code>	Σ	<code>\Omega</code>	Ω
<code>\Lambda</code>	Λ	<code>\Upsilon</code>	Υ		
<code>\varGamma</code>	\varGamma	<code>\varXi</code>	\varXi	<code>\varPhi</code>	\varPhi
<code>\varDelta</code>	\varDelta	<code>\varPi</code>	\varPi	<code>\varPsi</code>	\varPsi
<code>\varTheta</code>	\varTheta	<code>\varSigma</code>	\varSigma	<code>\varOmega</code>	\varOmega
<code>\varLambda</code>	\varLambda	<code>\varUpsilon</code>	\varUpsilon		

Table 7.2: Uppercase Greek letters

7.5 Delimiters

Delimiters are used to enclose some subformulas. In the following formula we use two delimiters: parentheses and square brackets: $[(a * b) + (c * d)]^2$; this typesets as $[(a * b) + (c * d)]^2$. \LaTeX knows that parentheses and square brackets are delimiters, and spaces them accordingly.

The standard delimiters are shown in Table 7.3. Note that delimiters are math symbols with special spacing rules, and you can use them in any way you please, not only in pairs. \LaTeX does not stop you from typing $\uparrow(x)$, which typesets as $\uparrow(x)$.

Name	Type	Typeset
left parenthesis	((
right parenthesis))
left bracket	[or \lbrack	[
right bracket] or \rbrack]
left brace	\{ or \lbrace	{
right brace	\} or \rbrace	}
backslash	\backslash	\
forward slash	/	/
left angle bracket	\langle	<
right angle bracket	\rangle	>
vertical line	or \vert	
double vertical line	\ or \Vert	
left floor	\lfloor	⌊
right floor	\rfloor	⌋
left ceiling	\lceil	⌈
right ceiling	\rceil	⌉
upward	\uparrow	↑
double upward	\Uparrow	⇑
downward	\downarrow	↓
double downward	\Downarrow	⇓
up-and-down	\updownarrow	↕
double up-and-down	\Updownarrow	⇕
upper-left corner	\ulcorner	⌠
upper-right corner	\urcorner	⌡
lower-left corner	\llcorner	⌢
lower-right corner	\lrcorner	⌣

Table 7.3: Standard delimiters.

Observe the difference in spacing between $||a||$ and $\|a\|$. The first, $||a||$, was typed incorrectly as $|| a ||$. As a result, the vertical bars are too far apart. The second was typed correctly using the appropriate delimiter commands: $\| a \|$. Here they

typesets as

$$\left[\sum_i a_i\right]^{1/p} \quad \left[\sum_i a_i\right]^{1/p}$$

You may prefer the second version with `\biggl[` and `\biggr]`.

Example 2. Groupings

```
\[
  \left( (a_1 b_1) - (a_2 b_2) \right) \right)
  \left( (a_2 b_1) + (a_1 b_2) \right) \right)
  \quad
  \bigl( (a_1 b_1) - (a_2 b_2) \bigr) \bigr)
  \bigl( (a_2 b_1) + (a_1 b_2) \bigr) \bigr)
\]
```

typesets as

$$((a_1 b_1) - (a_2 b_2)) ((a_2 b_1) + (a_1 b_2)) \quad ((a_1 b_1) - (a_2 b_2)) ((a_2 b_1) + (a_1 b_2))$$

You may prefer the clearer groupings provided by `\bigl(` and `\bigr)`.

Example 3. Inline formulas

```
\left\lvert \frac{b'}{d'} \right\rvert \right\lvert
```

Here the delimiters produced by `\left` and `\right` use too much interline space in $\left\lvert \frac{b'}{d'} \right\rvert$. Use `\bigl` and `\bigr` to produce delimiters that fit within the normal line spacing: $\bigl\lvert \frac{b'}{d'} \bigr\rvert$, typed as

```
\bigl\lvert \frac{b'}{d'} \bigr\rvert \bigr\lvert
```

Here are some variants: $\left\lvert \frac{b'}{d'} \right\rvert$, $\left\lvert \frac{b'}{d'} \right\rvert$, $\left\lvert \frac{b'}{d'} \right\rvert$, $\left\lvert \frac{b'}{d'} \right\rvert$

typed as

```
$\bigl\lvert \frac{b'}{d'} \bigr\rvert \bigr\lvert$,
\quad \Bigl\lvert \frac{b'}{d'} \Bigr\rvert
\Bigr\lvert \quad \biggl\lvert \frac{b'}{d'} \biggr\rvert \biggr\lvert
\quad \Biggl\lvert \frac{b'}{d'} \Biggr\rvert \Biggr\lvert$
```

7.5.4 Delimiters as binary relations

The symbol `|` can be used as a delimiter, as in $|x + y|$, and also as a binary relation, as in $\{x \in \mathcal{R} \mid x^2 \leq 2\}$. As a binary relation it is typed as `\mid`. The previous formula is typed as

```
\{ x \in \mathcal{R} \mid x^2 \leq 2 \}
```

`\bigm` and `\biggm` produce larger variants, with spacing on either side like binary relations. For example,

$$\left\{ x \mid \int_0^x t^2 dt \leq 5 \right\}$$

is typed as

```
\[
\left\{ x \biggm| \int_0^x t^2 dt \leq 5 \right\}
```

7.6 Operators

You cannot just type `\sin x` to typeset the sine function in math mode. Indeed,

```
\sin x
```

produces *sinx* instead of $\sin x$, as you intended. Type this function as

```
\sin x
```

The `\sin` command prints *sin* with the proper style and spacing. \LaTeX calls `\sin` an *operator* or *log-like function*.

7.6.1 Operator tables

There are two types of operators:

1. *Operators without limits*, such as `\sin`
2. *Operators with limits*, such as `\lim`, that take a subscript in inline mode and a “limit” in displayed math mode. For example, $\lim_{x \rightarrow 0} f(x) = 1$ is typed as

```
\lim_{x \to 0} f(x) = 1
```

The same formula displayed,

$$\lim_{x \rightarrow 0} f(x) = 1$$

is typed as


```
\[
  \lim_{x \to 0} f(x) = 1
\]
```

The operators are listed in Tables 7.4 and 7.5 (see also Appendix A.5). The entries in the last two rows of Table 7.5 can be illustrated by

$$\lim_{x \rightarrow 0} \quad \overline{\lim}_{x \rightarrow 0} \quad \lim_{x \rightarrow 0}^{\rightarrow} \quad \lim_{x \rightarrow 0}^{\leftarrow}$$

which are typed as

```
\[
  \varliminf_{x \to 0} \quad \quad \varlimsup_{x \to 0} \quad \quad
  \varinjlim_{x \to 0} \quad \quad \varprojlim_{x \to 0}
\]
```

Type	Typeset	Type	Typeset	Type	Typeset	Type	Typeset
<code>\arccos</code>	arccos	<code>\cot</code>	cot	<code>\hom</code>	hom	<code>\sin</code>	sin
<code>\arcsin</code>	arcsin	<code>\coth</code>	coth	<code>\ker</code>	ker	<code>\sinh</code>	sinh
<code>\arctan</code>	arctan	<code>\csc</code>	csc	<code>\lg</code>	lg	<code>\tan</code>	tan
<code>\arg</code>	arg	<code>\deg</code>	deg	<code>\ln</code>	ln	<code>\tanh</code>	tanh
<code>\cos</code>	cos	<code>\dim</code>	dim	<code>\log</code>	log		
<code>\cosh</code>	cosh	<code>\exp</code>	exp	<code>\sec</code>	sec		

Table 7.4: Operators without limits.

Type	Typeset	Type	Typeset
<code>\det</code>	det	<code>\limsup</code>	lim sup
<code>\gcd</code>	gcd	<code>\max</code>	max
<code>\inf</code>	inf	<code>\min</code>	min
<code>\lim</code>	lim	<code>\Pr</code>	Pr
<code>\liminf</code>	lim inf	<code>\sup</code>	sup
<code>\injlim</code>	inj lim	<code>\projlim</code>	proj lim
<code>\varliminf</code>	\varliminf	<code>\varlimsup</code>	$\overline{\lim}$
<code>\varinjlim</code>	\varinjlim	<code>\varprojlim</code>	\varprojlim

Table 7.5: Operators with limits.

The following examples illustrate some more entries from Table 7.5:

$\operatorname{inj\,lim}_{x\rightarrow 0}$

$\liminf_{x\rightarrow 0}$

$\limsup_{x\rightarrow 0}$

$\operatorname{proj\,lim}_{x\rightarrow 0}$

These operators were typed as

```
\[
\injlim_{x \to 0} \quad \liminf_{x \to 0} \quad
\limsup_{x \to 0} \quad \projlim_{x \to 0}
\]
```

You can force the limits in a displayed formula into the subscript position with the `\nolimits` command. For example, the formulas

$\operatorname{inj\,lim}_{x\rightarrow 0}$

$\liminf_{x\rightarrow 0}$

$\limsup_{x\rightarrow 0}$

$\operatorname{proj\,lim}_{x\rightarrow 0}$

are typed as

```
\[
\injlim\nolimits_{x \to 0} \quad
\liminf\nolimits_{x \to 0} \quad
\limsup\nolimits_{x \to 0} \quad
\projlim\nolimits_{x \to 0}
\]
```

7.6.2 Congruences

`\mod` is a special operator used for congruences. Congruences are usually typeset using the `\pmod` or `\pod` variant. There is also the `\bmod` command, which is used as a binary operation. All four commands are shown in Figure 7.6.

Type	Typeset
<code>\$a \equiv v \mod{\theta}\$</code>	$a \equiv v \mod \theta$
<code>\$a \bmod b\$</code>	$a \bmod b$
<code>\$a \equiv v \pmod{\theta}\$</code>	$a \equiv v \pmod{\theta}$
<code>\$a \equiv v \pod{\theta}\$</code>	$a \equiv v (\theta)$

Table 7.6: Congruences.

See Sections 15.1.2 and 15.1.9 for a discussion of related custom commands.

7.6.3 Large operators

Here is a sum typeset inline, $\sum_{i=1}^n x_i^2$, and displayed,

$$\sum_{i=1}^n x_i^2$$

In the latter form, the sum symbol is larger. Operators that behave in this way are called *large operators*. Table 7.7 gives a complete list of large operators.

You can use the `\nolimits` command if you wish to show the limits of large operators as subscripts and superscripts in a displayed math environment.

The formula

$$\bigsqcup_m X = a$$

is typed as

```
\[
\bigsqcup\nolimits_{\mathfrak{m}} X = a
\]
```

You can use the `\limits` command if you wish to show the limits of large operators below and above the operator symbol in an inline math environment. For example,

$\bigsqcup_m X = a$ is typed as

```
$\bigsqcup\limits_{\mathfrak{m}} X = a$
```

Sums and products are very important constructs. The examples

$$\frac{z^d - z_0^d}{z - z_0} = \sum_{k=1}^d z_0^{k-1} z^{d-k} \quad \text{and} \quad (T^n)'(x_0) = \prod_{k=0}^{n-1} T'(x_k)$$

are typed as

```
\[
\frac{z^{\mathfrak{d}} - z_{\mathfrak{0}}^{\mathfrak{d}}}{\mathfrak{z} - z_{\mathfrak{0}}} =
\sum_{k = 1}^{\mathfrak{d}} z_{\mathfrak{0}}^{\mathfrak{k} - 1} z^{\mathfrak{d} - \mathfrak{k}}
\text{\quad and \quad}
(T^{\mathfrak{n}})'(x_{\mathfrak{0}}) = \prod_{k=0}^{\mathfrak{n} - 1} T'(x_{\mathfrak{k}})
\]
```

7.6.4 Multiline subscripts and superscripts

The `\substack` command provides multiline limits for large operators. For instance,

$$\sum_{\substack{i < n \\ i \text{ even}}} x_i^2$$

is typed as

```
\[
\sum_{\substack{i < n \\ i \text{\text{ even}}}} x_{\mathfrak{i}}^2
\]
```

Type	Inline	Displayed
<code>\int_{a}^{b}</code>	\int_a^b	\int_a^b
<code>\oint_{a}^{b}</code>	\oint_a^b	\oint_a^b
<code>\iint_{a}^{b}</code>	\iint_a^b	\iint_a^b
<code>\iiint_{a}^{b}</code>	\iiint_a^b	\iiint_a^b
<code>\iiiint_{a}^{b}</code>	\iiiint_a^b	\iiiint_a^b
<code>\idotsint_{a}^{b}</code>	$\int \cdots \int_a^b$	$\int \cdots \int_a^b$
<code>\prod_{i=1}^n</code>	$\prod_{i=1}^n$	$\prod_{i=1}^n$
<code>\coprod_{i=1}^n</code>	$\coprod_{i=1}^n$	$\coprod_{i=1}^n$
<code>\bigcap_{i=1}^n</code>	$\bigcap_{i=1}^n$	$\bigcap_{i=1}^n$
<code>\bigcup_{i=1}^n</code>	$\bigcup_{i=1}^n$	$\bigcup_{i=1}^n$
<code>\bigwedge_{i=1}^n</code>	$\bigwedge_{i=1}^n$	$\bigwedge_{i=1}^n$
<code>\bigvee_{i=1}^n</code>	$\bigvee_{i=1}^n$	$\bigvee_{i=1}^n$
<code>\bigsqcup_{i=1}^n</code>	$\bigsqcup_{i=1}^n$	$\bigsqcup_{i=1}^n$
<code>\biguplus_{i=1}^n</code>	$\biguplus_{i=1}^n$	$\biguplus_{i=1}^n$
<code>\bigotimes_{i=1}^n</code>	$\bigotimes_{i=1}^n$	$\bigotimes_{i=1}^n$
<code>\bigoplus_{i=1}^n</code>	$\bigoplus_{i=1}^n$	$\bigoplus_{i=1}^n$
<code>\bigodot_{i=1}^n</code>	$\bigodot_{i=1}^n$	$\bigodot_{i=1}^n$
<code>\sum_{i=1}^n</code>	$\sum_{i=1}^n$	$\sum_{i=1}^n$

Table 7.7: Large operators.

		amsxtra	
Type	Typeset	Type	Typeset
<code>\acute{a}</code>	\acute{a}		
<code>\bar{a}</code>	\bar{a}		
<code>\breve{a}</code>	\breve{a}	<code>\spbreve</code>	\spbreve
<code>\check{a}</code>	\check{a}	<code>\spcheck</code>	\spcheck
<code>\dot{a}</code>	\dot{a}	<code>\spdot</code>	\spdot
<code>\ddot{a}</code>	\ddot{a}	<code>\spddot</code>	\spddot
<code>\dddota</code>	\dddota	<code>\spdddot</code>	\spdddot
<code>\grave{a}</code>	\grave{a}		
<code>\hat{a}</code>	\hat{a}		
<code>\widehat{a}</code>	\widehat{a}	<code>\sphat</code>	\sphat
<code>\mathring{a}</code>	\mathring{a}		
<code>\tilde{a}</code>	\tilde{a}		
<code>\widetilde{a}</code>	\widetilde{a}	<code>\sptilde</code>	\sptilde
<code>\vec{a}</code>	\vec{a}		

Table 7.8: Math accents

There is only one rule to remember.

Use the line separator command `\`.

You can use the `\substack` command wherever subscripts or superscripts are used. The lines are centered by `\substack`, so if you want them set flush left, as in

$$\sum_{\substack{i < n \\ i \text{ even}}} x_i^2$$

then use the `subarray` environment with the argument 1:

```
\[
\sum_{\begin{subarray}{l}
i < n\\
i \text{ even}
\end{subarray}}
x_{i}^2
\]
```

See Section 15.1.6 for another example.

7.7 Math accents

The accents used in text (see Section 3.4.7) cannot be used in math formulas. A separate set of commands is provided for accents in formulas. All math accents are shown in Table 7.8 (see also Appendix A.7). The `amsxtra` package is needed for the accents in the second column. To use them, make sure to place in the preamble the line

```
\usepackage{amsxtra}
```

You can also use double accents, such as

```
\[
  \hat{\hat{A}}
\]
```

which typesets as $\hat{\hat{A}}$.

The two “wide” varieties, `\widehat` and `\widetilde`, expand to fit the symbols (their arguments) covered: \hat{A} , \hat{ab} , \widehat{iii} , $\widehat{ai ai}$, \widehat{iiii} , and \tilde{A} , \tilde{ab} , \widetilde{iii} , $\widetilde{ai ai}$, \widetilde{iiii} (the last example is typed as `\widetilde{iiii}`). If the base is too wide, the accent is centered:

$$\widehat{ABCDE}$$

The “sp” commands, provided by the `amsxtra` package, are used for superscripts, as illustrated in Table 7.8. If you use a lot of accented characters, you should appreciate custom commands (see Section 15.1.1).

Notice the difference between \bar{a} and \overline{a} , typed as

```
\bar{a} \overline{a}
```

For other examples of the `\overline` command; see Section 7.8.2.

To use an arbitrary symbol as an accent or to create “underaccents,” use Javier Bezos’ `accents` package.

7.8 Stretchable horizontal lines

L^AT_EX provides three types of stretchable horizontal lines that appear above or below a formula: braces, bars, and arrows. There are also stretchable arrow math symbols.

7.8.1 Horizontal braces

The `\overbrace` command places a brace of variable size above its argument, as in

$$\overbrace{a + b + \cdots + z}$$

which is typed as

```
\[
  \overbrace{a + b + \dots + z}
\]
```

A superscript adds a label to the brace, as in

$$\overbrace{a + a + \cdots + a}^n$$

which is typed as

```
\[
  \overbrace{a + a + \dots + a}^{\text{\texttt{\char"007Bn}}}
\]
```

The `\underbrace` command works similarly, placing a brace below its argument. A subscript adds a label to the brace, as in

$$\underbrace{a + a + \cdots + a}_n$$

which is typed as

```
\[
  \underbrace{a + a + \dots + a}_{\text{\texttt{\char"007Bn}}}
\]
```

The following example combines these two commands:

$$\underbrace{\overbrace{a + \cdots + a}^{(m-n)/2} + \underbrace{b + \cdots + b}_n + \overbrace{a + \cdots + a}^{(m-n)/2}}_m$$

This example is typed as

```
\[
  \underbrace{
    \overbrace{a + \dots + a}^{\text{\texttt{\char"007B(m - n)/2}}}
    + \underbrace{b + \dots + b}_{\text{\texttt{\char"007Bn}}}
    + \overbrace{a + \dots + a}^{\text{\texttt{\char"007B(m - n)/2}}}
  }_{\text{\texttt{\char"007Bm}}}
\]
```

7.8.2 Overlines and underlines

The `\overline` and `\underline` commands draw lines above or below a formula. For example,

$$\overline{\overline{X \cup \overline{X}}} = \overline{X}$$

is typed as

```
\[
\overline{ \overline{X} \cup \overline{\overline{X}} }
= \overline{ \overline{X} }
\]
```

Similarly, you can place arrows above and below an expression:

$$\begin{array}{cccc} \overleftarrow{a} & \overrightarrow{aa} & & \\ \overleftrightarrow{aaa} & \overleftarrow{aaaa} & \overrightarrow{aaaaa} & \overleftrightarrow{aaaaaa} \end{array}$$

which is typed as

```
\begin{gather*}
\overleftarrow{a} \quad \quad \quad \overrightarrow{aa} \\
\overleftrightharpoonup{aaa} \quad \quad \quad \underleftarrow{aaaa} \quad \quad \quad \underrightarrow{aaaaa} \\
\underrightarrow{aaaaa} \quad \quad \quad \underleftrightharpoonup{aaaaaa}
\end{gather*}
```

7.8.3 Stretchable arrow math symbols

There are two stretchable arrow math symbols that extend to accommodate a formula above or below the arrows with the `\xleftarrow` and `\xrightarrow` commands. The formula on top is given as the argument (possibly empty) and the formula below is an optional argument.

$$A \xrightarrow{1-1} B \xleftarrow[\alpha \rightarrow \beta]{\text{onto}} C \xleftarrow[\gamma]{} D \leftarrow E$$

is typed as

```
\[
A \xrightarrow{\text{1-1}} B \xleftarrow[\alpha \rightarrow \beta]{\text{onto}} C \xleftarrow[\gamma]{} D \leftarrow E
\]
```

There are other stretchable arrow math symbols described in Section 9.8, but they can only be used in commutative diagrams. See Section 14.1.2 for the `TikZ` arrows and the `tikz-cd` package arrows.

7.9 Building a formula step-by-step

It is easy to build complex formulas from the components described in this section. Try the formula, it looks pretty frightening:

$$\sum_{i=1}^{\left\lceil \frac{n}{2} \right\rceil} \left(x_{i,i+1}^{i^2} \right) \frac{\sqrt{\mu(i)^{\frac{3}{2}}(i^2-1)}}{\sqrt[3]{\rho(i)-2} + \sqrt[3]{\rho(i)-1}}$$

We build this formula in several steps. Create a new file in your **work** folder. Name it **formula.tex**, type in the following lines, and save it:

```
%File: formula.tex
\documentclass{sample}
\begin{document}
\end{document}
```

At present, the file has an empty **document** environment. Type each part of the formula as an inline or displayed formula within this environment, so that you can typeset the document and check for errors.

Step 1 We start with $\left\lceil \frac{n}{2} \right\rceil$. Type the following line into **formula.tex**:

```
$\left[ \frac{n}{2} \right]$
```

and test it by typesetting the document.

Step 2 Now you can do the sum

$$\sum_{i=1}^{\left\lceil \frac{n}{2} \right\rceil}$$

For the superscript, you can copy and paste the formula created in Step 1 (without the dollar signs), so that you have

```
\[
\sum_{i = 1}^{\left[ \frac{n}{2} \right]}
\]
```

Step 3 Next, do the two formulas in the binomial

$$x_{i,i+1}^{i^2} \quad \left[\frac{i+3}{3} \right]$$

Type them as separate formulas in **formula.tex**:

```
\[
x_{i, i + 1}^{i^2} \quad \left[ \frac{i + 3}{3} \right]
\]
```

Step 4 Now it is easy to do the binomial. Piece together the following formula by copying and pasting the previous formulas and dropping the `\quad` command:

```
\[
  \binom{x_{i,i+1}^{i^2}}{\left[\frac{i+3}{3}\right]}
\]
```

which typesets as

$$\binom{x_{i,i+1}^{i^2}}{\left[\frac{i+3}{3}\right]}$$

Step 5 Next, type the formula under the square root, $\mu(i)^{\frac{3}{2}}(i^2 - 1)$:

```
\mu(i)^{\frac{3}{2}} (i^2 - 1)$
```

and then the square root, $\sqrt{\mu(i)^{\frac{3}{2}}(i^2 - 1)}$:

```
\sqrt{\mu(i)^{\frac{3}{2}} (i^2 - 1)}$
```

Step 6 The two cube roots, $\sqrt[3]{\rho(i) - 2}$ and $\sqrt[3]{\rho(i) - 1}$, are easy to type:

```
\sqrt[3]{\rho(i) - 2}$ \sqrt[3]{\rho(i) - 1}$
```

Step 7 Now the fraction

$$\frac{\sqrt{\mu(i)^{\frac{3}{2}}(i^2 - 1)}}{\sqrt[3]{\rho(i) - 2} + \sqrt[3]{\rho(i) - 1}}$$

is typed, copied, and pasted together as

```
\[
  \frac{\sqrt{\mu(i)^{\frac{3}{2}} (i^2 - 1)}}{\sqrt[3]{\rho(i) - 2} + \sqrt[3]{\rho(i) - 1}}
\]
```

Step 8 Finally, the whole formula,

$$\sum_{i=1}^{\left[\frac{n}{2}\right]} \binom{x_{i,i+1}^{i^2}}{\left[\frac{i+3}{3}\right]} \frac{\sqrt{\mu(i)^{\frac{3}{2}}(i^2 - 1)}}{\sqrt[3]{\rho(i) - 2} + \sqrt[3]{\rho(i) - 1}}$$

is formed by copying and pasting the pieces together, leaving only one pair of displayed math delimiters:

```

\[
  \sum_{i = 1}^{\left[ \frac{n}{2} \right]}
    \binom{x_{i, i + 1}^{i^2}}
      { \left[ \frac{i + 3}{3} \right] }
    \frac{\sqrt{\mu(i)^{\frac{3}{2}}} (i^2 - 1) }
      {\sqrt[3]{\rho(i)-2} + \sqrt[3]{\rho(i) - 1}}
\]

```

Note the use of:

- hierarchical indentation, to keep track of the structure of the formula;
- spacing to help highlight the braces—some text editors help you balance braces;
- separate lines for the various pieces of formulas that are more than a line long.



Tip Keep your source file readable.

L^AT_EX does not care how its input is formatted, and would happily accept the following:

```

\[ \sum_{i=1}^{\left[ \frac{n}{2} \right]} \binom{x_{i,i+1}^{i^2}}
{\left[ \frac{i+3}{3} \right]} \frac{\sqrt{\mu(i)^{\frac{3}{2}}} (i^2-1)}
{\sqrt[3]{\rho(i)-2} + \sqrt[3]{\rho(i)-1}} \]

```

But this haphazard style not only makes it more difficult for your coauthors or editor to work with your source file, it also makes finding mistakes difficult. Try to find the error in the next version:

```

\[ \sum_{i=1}^{\left[ \frac{n}{2} \right]}
\binom{x_{i,i+1}^{i^2}}{\left[ \frac{i+3}{3} \right]}
\frac{\sqrt{\mu(i)^{\frac{3}{2}}} (i^2-1)}{\sqrt[3]{\rho(i)-2} + \sqrt[3]{\rho(i)-1}} \]

```

Answer: `\frac{3}{2}` should be followed by `}}` and not by `}}`.

7.10 Formula Gallery

In this section, I present a collection of formulas—some simple, some complex—that illustrate the power of L^AT_EX.

Some of these examples require the `amssymb` package, so it is a good idea to include the line

`\usepackage{amssymb,latexsym}`

following the `\documentclass` line of any article.

Formula 1 A set-valued function

$$x \mapsto \{c \in C \mid c \leq x\}$$

```
\[
  x \mapsto \{ c \in C \mid c \leq x \}
\]
```

Formula 2

$$\left| \bigcup (I_j \mid j \in J) \right| < m$$

```
\[
  \left| \bigcup ( I_{\{j\}} \mid j \in J ) \right| < \mathfrak{m}
\]
```

We use the delimiters `\left|` and `\right|`, see Section 7.5.1. The Fraktur `m` is introduced in Section 8.4.2.

Formula 3 Note that you have to add spacing both before and after the text fragment in the following example. The argument of `\text` is typeset in text mode, so spaces are recognized.

$$A = \{x \in X \mid x \in X_i \text{ for some } i \in I\}$$

```
\[
  A = \{ x \in X \mid x \in X_{\{i\}},
        \text{ for some } i \in I \}
\]
```

Formula 4 Spaces to show logical structure:

$$\langle a_1, a_2 \rangle \leq \langle a'_1, a'_2 \rangle \quad \text{iff} \quad a_1 < a'_1 \quad \text{or} \quad a_1 = a'_1 \text{ and } a_2 \leq a'_2$$

```
\[
  \langle a_{\{1\}}, a_{\{2\}} \rangle \leq
  \langle a'_{\{1\}}, a'_{\{2\}} \rangle \quad \text{iff}
  \quad \langle a_{\{1\}} < a'_{\{1\}} \rangle \quad \text{or}
  \quad \langle a_{\{1\}} = a'_{\{1\}} \rangle \text{ and } a_{\{2\}} \leq a'_{\{2\}}
\]
```

Note that in `iff` (in the argument of the first `\text`) the second `f` is enclosed in braces to avoid the use of the ligature—the merging of the two `f`'s. For the proper way of typesetting `iff` without a ligature; see Section 3.4.6.

Formula 5 Here are some examples of Greek letters:

$$\Gamma_{u'} = \{\gamma \mid \gamma < 2\chi, B_\alpha \not\subseteq u', B_\gamma \subseteq u'\}$$

```
\[
\Gamma_{u'} = \{\gamma \mid \gamma < 2\chi, B_{\alpha}
\subseteq u', B_{\gamma} \subseteq u'\}
\]
```

See Section A.1 for a complete listing of Greek letters. We use the command `\sq` to properly space the formula. This command can be used both in text and in math.

Formula 6 `\mathbb` allows you to use the blackboard bold math alphabet, which only provides capital letters:

$$A = B^2 \times \mathbb{Z}$$

```
\[
A = B^2 \times \mathbb{Z}
\]
```

Formula 7 `\left[` and `\right]` provide stretched delimiters:

$$y^C \equiv z \vee \bigvee_{i \in C} [s_i^C] \pmod{\Phi}$$

```
\[
y^C \equiv z \vee \bigvee_{i \in C} \left[ s_i^C \right]
\pmod{\Phi}
\]
```

Notice how the superscript is set directly above the subscript in s_i^C .

Formula 8 A complicated congruence:

$$y \vee \bigvee ([B_\gamma] \mid \gamma \in \Gamma) \equiv z \vee \bigvee ([B_\gamma] \mid \gamma \in \Gamma) \pmod{\Phi^x}$$

```
\[
y \vee \bigvee ( [B_{\gamma}] \mid \gamma
\in \Gamma ) \equiv z \vee \bigvee ( [B_{\gamma}]
\mid \gamma \in \Gamma ) \pmod{\Phi^x}
\]
```

Formula 9 Use `\nolimits` to force the “limit” of the large operator to display as a subscript (see Section 7.6.3):

$$f(\mathbf{x}) = \bigvee_{\mathbf{m}} \left(\bigwedge_{\mathbf{m}} (x_j \mid j \in I_i) \mid i < \aleph_{\alpha} \right)$$

```
\[
  f(\mathbf{x}) =
    \bigvee\nolimits_{\!\!\!\mathfrak{m}}
    \left(
      \bigwedge\nolimits_{\mathfrak{m}}
      ( x_{\mathfrak{j}} \mid j \in I_{\mathfrak{i}} )
      \mid i < \aleph_{\alpha}
    \right)
\]
```

Notice that I inserted a negative space (`\!`) to bring the `\mathfrak{m}` a little closer to the big join symbol \bigvee .

Formula 10 The `\left.` command gives a blank left delimiter, which is needed to balance the `\right|` command:

$$\widehat{F}(x) \Big|_a^b = \widehat{F}(b) - \widehat{F}(a)$$

```
\[
  \left. \widehat{F}(x) \right|_{\mathfrak{a}}^{\mathfrak{b}}
  = \widehat{F}(b) - \widehat{F}(a)
\]
```

Formula 11 The `\underset` and `\overset` commands build new symbols (see Section 8.3.1):

$$u +_{\alpha} v \overset{1}{\sim} w \overset{2}{\sim} z$$

```
\[
  u \underset{\alpha}{+} v \overset{1}{\thicksim} w
    \overset{2}{\thicksim} z
\]
```

Note that the new symbols $\overset{1}{\sim}$ and $\overset{2}{\sim}$ are binary relations and $+_{\alpha}$ is a binary operation.

Formula 12 Small size bold `\def`:

$$f(x) \stackrel{\text{def}}{=} x^2 - 1$$

```
\[
  f(x) \overset{\mathbf{def}}{=} x^2 - 1
\]
```

Formula 13 Math accents run amok:

$$\overbrace{a^{\breve{\vee}} + b^{\breve{\vee}} + \dots + z^{\breve{\vee}}}^{\breve{\breve{n}}}$$

```
\[
  \overbrace{a\spcheck + b\spcheck + \dots + z\spcheck}^
{\breve{\breve{n}}}
\]
```

Recall that for the `\sp` commands you need the `amsxtra` package.

Formula 14

$$\left| \begin{array}{cc} a + b + c & uv \\ a + b & c + d \end{array} \right| = 7$$

```
\[
  \begin{vmatrix}
    a + b + c & uv \\
    a + b & c + d
  \end{vmatrix}
  = 7
\]
```

$$\left\| \begin{array}{cc} a + b + c & uv \\ a + b & c + d \end{array} \right\| = 7$$

```
\[
  \begin{Vmatrix}
    a + b + c & uv \\
    a + b & c + d
  \end{Vmatrix}
  = 7
\]
```

Formula 15

$$\alpha^2 \sum_{j \in \mathbf{N}} b_{ij} \hat{y}_j = \sum_{j \in \mathbf{N}} b_{ij}^{(\lambda)} \hat{y}_j + (b_{ii} - \lambda_i) \hat{y}_i \hat{y}$$

```
\[
  \boldsymbol{\alpha}^2 \sum_{j \in \mathbf{N}} b_{ij}
  \hat{y}_{-j} = \sum_{j \in \mathbf{N}} b_{ij}^{(\lambda)} \hat{y}_{-j}
  + (b_{ii} - \lambda_i) \hat{y}_{-i} \hat{y}
\]
```

`\mathbf{N}` makes a bold \mathbf{N} and `\boldsymbol{\alpha}` produces a bold $\boldsymbol{\alpha}$ (see Section 8.4.2).

Formula 16 To produce the formula

$$\left(\prod_{j=1}^n \hat{x}_j\right) H_c = \frac{1}{2} \hat{k}_{ij} \det \hat{\mathbf{K}}(i|i)$$

try typing

```
\[
\left( \prod_{j = 1}^n \hat{x}_{j} \right) H_{c} =
\frac{1}{2} \hat{k}_{ij} \det \hat{\mathbf{K}}(i|i)
\]
```

which typesets as

$$\left(\prod_{j=1}^n \hat{x}_j\right) H_c = \frac{1}{2} \hat{k}_{ij} \det \hat{\mathbf{K}}(i|i)$$

This is not quite right. You can correct the overly large parentheses by using the `\biggl` and `\biggr` commands in place of `\left(` and `\right)`, respectively (see Section 7.5.2). Adjust the small hat over \mathbf{K} by using `\widehat`:

```
\[
\biggl( \prod_{j = 1}^n \hat{x}_{j} \biggr)
H_{c} = \frac{1}{2} \hat{k}_{ij}
\det \widehat{\mathbf{K}}(i|i)
\]
```

which gives you the desired formula.

Formula 17 In this formula, I have used `\overline{I}` to get \bar{I} . You could, instead, use `\bar{I}`, which is typeset as \bar{I} .

$$\det \mathbf{K}(t = 1, t_1, \dots, t_n) = \sum_{I \in \mathbf{n}} (-1)^{|I|} \prod_{i \in I} t_i \prod_{j \in I} (D_j + \lambda_j t_j) \det \mathbf{A}^{(\lambda)}(\bar{I}|\bar{I}) = 0$$

```
\[
\det \mathbf{K}(t = 1, t_{1}, \dots, t_{n}) =
\sum_{I \in \mathbf{n}} (-1)^{|I|} \prod_{i \in I} t_{i}
\prod_{j \in I} (D_{j} + \lambda_{j} t_{j})
\det \mathbf{A}^{(\lambda)}
(\overline{I} | \overline{I}) = 0
\]
```

Formula 18 The command `\|` provides the $\|$ math symbol in this formula:

$$\lim_{(v,v') \rightarrow (0,0)} \frac{H(z+v) - H(z+v') - BH(z)(v-v')}{\|v-v'\|} = 0$$


```
\[
\lim_{(v, v') \rightarrow (0, 0)}
\frac{H(z + v) - H(z + v') - BH(z)(v - v')}{\|v - v'\|} = 0
\]
```

Formula 19 This formula uses the calligraphic math alphabet (introduced in Section 8.4.2):

$$\int_{\mathcal{D}} |\overline{\partial} u|^2 \Phi_0(z) e^{\alpha|z|^2} \geq c_4 \alpha \int_{\mathcal{D}} |u|^2 \Phi_0 e^{\alpha|z|^2} + c_5 \delta^{-2} \int_A |u|^2 \Phi_0 e^{\alpha|z|^2}$$

```
\[
\int_{\mathcal{D}} |\overline{\partial} u|^2 \Phi_0(z) e^{\alpha|z|^2}
\geq c_4 \alpha \int_{\mathcal{D}} |u|^2 \Phi_0 e^{\alpha|z|^2}
+ c_5 \delta^{-2} \int_A |u|^2 \Phi_0 e^{\alpha|z|^2}
\]
```

Formula 20 The `\hdotsfor` command sets dots that span multiple columns in a matrix. The `\dfraction` command is the displayed variant of the `\frac` command (see Section 7.4.1), used here because the matrix entries with `\frac` would look too small.

$$\mathbf{A} = \begin{pmatrix} \frac{\varphi \cdot X_{n,1}}{\varphi_1 \times \varepsilon_1} & (x + \varepsilon_2)^2 & \cdots & (x + \varepsilon_{n-1})^{n-1} & (x + \varepsilon_n)^n \\ \frac{\varphi \cdot X_{n,1}}{\varphi_2 \times \varepsilon_1} & \frac{\varphi \cdot X_{n,2}}{\varphi_2 \times \varepsilon_2} & \cdots & (x + \varepsilon_{n-1})^{n-1} & (x + \varepsilon_n)^n \\ \hdotsfor{5} \\ \frac{\varphi \cdot X_{n,1}}{\varphi_n \times \varepsilon_1} & \frac{\varphi \cdot X_{n,2}}{\varphi_n \times \varepsilon_2} & \cdots & \frac{\varphi \cdot X_{n,n-1}}{\varphi_n \times \varepsilon_{n-1}} & \frac{\varphi \cdot X_{n,n}}{\varphi_n \times \varepsilon_n} \end{pmatrix} + \mathbf{I}_n$$

```
\[
\mathbf{A} =
\begin{pmatrix}
\frac{\varphi \cdot X_{n,1}}{\varphi_1 \times \varepsilon_1} & (x + \varepsilon_2)^2 & \cdots & (x + \varepsilon_{n-1})^{n-1} & (x + \varepsilon_n)^n \\
\frac{\varphi \cdot X_{n,1}}{\varphi_2 \times \varepsilon_1} & \frac{\varphi \cdot X_{n,2}}{\varphi_2 \times \varepsilon_2} & \cdots & (x + \varepsilon_{n-1})^{n-1} & (x + \varepsilon_n)^n \\
\hdotsfor{5} \\
\frac{\varphi \cdot X_{n,1}}{\varphi_n \times \varepsilon_1} & \frac{\varphi \cdot X_{n,2}}{\varphi_n \times \varepsilon_2} & \cdots & \frac{\varphi \cdot X_{n,n-1}}{\varphi_n \times \varepsilon_{n-1}} & \frac{\varphi \cdot X_{n,n}}{\varphi_n \times \varepsilon_n}
\end{pmatrix} + \mathbf{I}_n
\]
```

```

{\varphi_{2} \times \varepsilon_{2}} & \cdots &
(x + \varepsilon_{n-1})^{n-1}
& (x + \varepsilon_n)^n \\
\hdotsfor{5} \\
\dfrac{\varphi \cdot X_{n,1}}{\varepsilon_1} & \dfrac{\varphi \cdot X_{n,2}}{\varepsilon_2} & \cdots
& \dfrac{\varphi \cdot X_{n,n-1}}{\varepsilon_{n-1}} & \dfrac{\varphi \cdot X_{n,n}}{\varepsilon_n} \\
& \varphi \times \varepsilon_n \\
\end{pmatrix}
+ \mathbf{I}_n
\]

```

Recall the discussion of `\dots` vs. `\cdots` and `\ldots` in Section 7.4.3. In this formula, we have to use `\cdots`. Matrices are discussed in detail in Section 9.7.1.

Note the use of the command `\[10pt]`. If you use `\[` instead, the first and second lines of the matrix are set too close.

I show you in Section 15.1.2 how to rewrite this formula to make it shorter and more readable.

A note on the `latexsym` package. As L^AT_EX has evolved, many of the symbols from the `latexsym` package have been incorporated into other packages such as `amssymb`, which provides a broader range of mathematical symbols. In modern documents, it's more common to use the `amssymb` package for a comprehensive set of mathematical symbols, but `latexsym` remains a part of the L^AT_EX core for compatibility and for cases where only its specific set of symbols is needed.



More math

In the previous chapter, we discussed the basic building blocks of a formula and how to put them together to form more complex formulas. This chapter starts out by going one step lower, to the characters that make up a formula. We discuss math symbols and math alphabets.

L^AT_EX was designed for typesetting math, so it is not surprising that it contains a very large number of math symbols. Section 8.1 classifies and describes them. Appendix A lists them all. Section 8.2 introduces the STIX symbols, some 2,000 of them. Section 8.3 discusses how to build new symbols from existing ones. Math alphabets and symbols are discussed in Section 8.4. Horizontal spacing commands in math are described in Section 8.5.

L^AT_EX provides a variety of ways to number and tag equations. These techniques are described in Section 8.6. We conclude the chapter with two minor topics: generalized fractions and boxed formulas.

8.1 Spacing of symbols

L^AT_EX provides a large variety of math symbols: Greek characters (α), binary operations (\circ), binary relations (\leq), negated binary relations ($\not\leq$), arrows (\nearrow), delim-

iters ({), and so on. Consider the formula

$$A = \{x \in X \mid x\beta \geq xy > (x+1)^2 - \alpha\}$$

which is typed as

```
\[
  A = \{ x \in X \mid x \beta \geq x y
    > (x + 1)^{2} - \alpha \}
\]
```

The spacing of the symbols in the formula varies. In $x\beta$, the two symbols are very close. In $x \in X$, there is some space around the \in , and in $x + 1$, there is somewhat less space around the $+$.

All the math symbols provided by L^AT_EX are listed in the tables of Appendix A.

8.1.1 Classification

L^AT_EX classifies symbols into several categories or *types* and spaces them accordingly. In the formula

$$A = \{x \in X \mid x\beta \geq xy > (x+1)^2 - \alpha\}$$

we find:

- ordinary math symbols: A , x , X , β , and so on;
- binary relations: $=$, \in , $|$, \geq , and $>$;
- binary operations: $+$ and $-$;
- delimiters: $\{$, $\}$, $($, and $)$.

As a rule, you do not have to be concerned with whether or not a given symbol in a formula, say \times , is a binary operation. L^AT_EX knows and spaces the typeset symbol correctly, unless, the symbol is invoked by a custom command.

8.1.2 Three exceptions

There are three symbols with more than one classification:

$$+ \quad - \quad |$$

$+$ or $-$ could be either a binary operation, for instance, $a - b$, or a sign, for instance, $-b$.



Rule: $+$ and $-$

$+$ or $-$ are binary operations when preceded and followed by a symbol or an empty group, $\{ \}$.

So, for instance, in

$$\begin{aligned}(A + BC)x + Cy &= 0, \\ Ex + (F + G)y &= 23.\end{aligned}$$

which is typed as (see the `alignat*` environment in Section 9.5.4)

```
\begin{alignat*}{2}
  (A + B C)x &+{}+{} &C &\phantom{0} &y = 0, \\
  Ex &+{}+{} &(F + G)&y = 23.
\end{alignat*}
```

we use the empty groups, `{ }`, to tell L^AT_EX that the second `+` in line 1 and the first `+` in line 2 of the formula are binary operations. If we leave out the empty groups, and type instead

```
\begin{alignat*}{2}
  (A + B C)x &+ &C &\phantom{0} &y = 0, \\
  Ex &+ &(F + G)&y = 23.
\end{alignat*}
```

we get

$$\begin{aligned}(A + BC)x + Cy &= 0, \\ Ex + (F + G)y &= 23.\end{aligned}$$

Another way to achieve the proper spacing is by using the `\phantom` command; see Section 8.1.5. This problem (getting the proper classification) often arises in split formulas, for example if the formula is split just before a `+` or `-`, you should start the next line with `{ }+` or `{ }-`. See Section 9.3 for examples.

The `|` symbol can play several different roles in a math formula, so L^AT_EX provides separate commands to specify the meaning of the symbol.



Rule: The four roles of the `|` symbol

- `|` ordinary math symbol,
- `\mid` binary relation,
- `\left|` left delimiter,
- `\right|` right delimiter.

Note the differences between the spacing in $a|b$, typed as `$a | b$`, and in $a \mid b$, typed as `$a \mid b$`.

Name	Width	Short	Long
1 mu (math unit)	ı		<code>\mspace{1mu}</code>
thinspace	ıı	<code>\,</code>	<code>\thinspace</code>
medspace	ııı	<code>\:</code>	<code>\medspace</code>
thickspace	ıııı	<code>\;</code>	<code>\thickspace</code>
interword space	ııııı	<code>\ı</code>	
1 em	ıııııı		<code>\quad</code>
2 em	ıııııııı		<code>\qquad</code>
Negative space			
1 mu	ı		<code>\mspace{-1mu}</code>
thinspace	ıı	<code>\!</code>	<code>\negthinspace</code>
medspace	ııı		<code>\negmedspace</code>
thickspace	ıııı		<code>\negthickspace</code>

Table 8.1: Math spacing commands.

8.1.3 Spacing commands

There are some situations where \LaTeX cannot typeset a formula properly and you have to add spacing commands. Luckily, \LaTeX provides a variety of spacing commands, listed in Table 8.1. The `\neg` commands remove space by “reversing the print head”.

The `\quad` and `\qquad` commands are often used to adjust aligned formulas (see Chapter 9) or to add space before text in a math formula. The size of `\quad` (= 1 em) and `\qquad` (= 2 em) depends on the current font.

The `\,` and `\!` commands are the most useful for fine tuning math formulas; see some examples in the *Formula Gallery* and in the next section. The `\mspace` command and the math unit *mu* provides you with even finer control. 18 mu = 1 em, defined in Section 3.7.3. For example, `\mspace{3mu}` adds a space that is 1/6 em long. There is an interesting use of mu on page 90.

8.1.4 Examples

We present some examples of fine tuning. One more example can be found in Section 8.3.1.

Example 1 In Section 1.6, we type the formula $\int_0^\pi \sin x \, dx = 2$ as

```
\int_{0}^{\pi} \sin x \, dx = 2
```

Notice the `thinspace` spacing command `\,` between `\sin x` and `dx`. Without the command, \LaTeX would have crowded $\sin x$ and dx : $\int_0^\pi \sin x dx = 2$.

Example 2 $|-f(x)|$, typed as `$|-f(x)|$`, is spaced incorrectly. $-$ becomes a binary operation by the `+` and $-$ rule. To get the correct spacing, as in $|-f(x)|$, type `$\left|-f(x)\right|$`. This form tells L^AT_EX that the first $|$ is a left delimiter, by the `|` rule, and therefore $-$ is the unary minus sign, not the binary subtraction operation.

Example 3 In $\sqrt{5}\text{side}$, typed as

```
$\sqrt{5} \text{\texttt{\text{side}}}$
```

$\sqrt{5}$ is too close to side. So type it as

```
$\sqrt{5} \text{\texttt{\text{side}}} \text{\texttt{\text{side}}}$
```

which typesets as $\sqrt{5}\text{side}$.

Example 4 In $\sin x / \log n$, the division symbol $/$ is too far from $\log n$, so type

```
$\sin x / \! \log n$
```

which prints $\sin x / \log n$.

Example 5 In $f(1/\sqrt{n})$, typed as

```
$f(1 / \sqrt{n})$
```

the square root almost touches the closing parenthesis. To correct it, type

```
$f(1 / \sqrt{n}\,)$
```

which typesets as $f(1/\sqrt{n})$.

There is one more symbol with special spacing: the `\colon` command, used for formulas such as $f: A \rightarrow B$ (typed as `$f \colon A \to B$`). Observe that `$f: A \to B$` typesets as $f : A \rightarrow B$. The spacing is awful. See Section 8.3.3 on how to declare the type of a symbol.

8.1.5 The phantom command

The `` command (introduced for text in Section 3.7.1) produces a space in a formula equivalent to the space that would be occupied by its typeset argument. This command is one of the most powerful tools available to us for fine tuning alignments. Here are two simple illustrations:

$$A = \begin{pmatrix} 1 & 3 & 1 \\ 2 & 1 & 1 \\ -2 & 2 & -1 \end{pmatrix}$$

typed as

```
\[
A = \begin{pmatrix}
\phantom{-}1 & \phantom{-}3 & \phantom{-}1\\
\phantom{-}2 & \phantom{-}1 & \phantom{-}1\\
-2 & \phantom{-}2 & -1
\end{pmatrix}
\]
```

and

$$\begin{aligned}a + b + c + d &= 0, \\ c + d + e &= 5.\end{aligned}$$

typed as

```
\begin{align*}
a + b + c &+ d \phantom{+e} = 0, \\
c &+ d + e = 5.
\end{align*}
```

Note that `` yields incorrect spacing by the + and – rule:

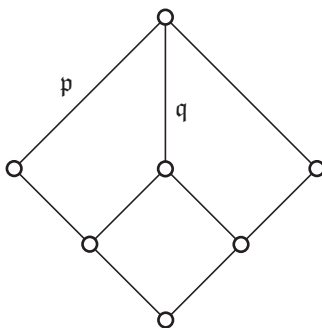
$$\begin{aligned}a + b + c + d &= 0, \\ c + d + e &= 5.\end{aligned}$$

See Section 9.6.2 for an additional example.

8.2 The STIX math symbols

Swinging it

In a paper of mine (see arXiv:1312.2537), I introduce the concept of a *swing*: a prime interval p swings to another one, q , as exemplified by this diagram:



L^AT_EX provides a nice curved arrow, `\curvearrowright`, typeset as \curvearrowright ; unfortunately, it is upside down (you don’t swing that way). Utilizing the `graphicx` package, I solved my problem by defining


```
\newcommand{\swing}
{\mathbin{\{\rotatebox{180}{\curvearrowleft}\}}}
```

which turns `\curvearrowleft`, ↶, 180 degrees: $p \curvearrowleft q$. Raise it with `\raisebox`: $p \overset{\curvearrowleft}{\curvearrowleft} q$, typed as

```
\mathfrak p \swingraised \mathfrak q
```

where

```
\newcommand{\swingraised}{\mathbin{\raisebox{3.0pt}
{\rotatebox{160}{\curvearrowleft}\}}}
```

A better way to solve the problem is by utilizing the 2,000 or so new math symbols offered by STIX.

The STIX project

“The mission of the Scientific and Technical Information Exchange (STIX) font creation project is the preparation of a comprehensive set of fonts that serve the scientific and engineering community in the process from manuscript creation through final publication, both in electronic and print formats”. The symbols were completed by 2006. The L^AT_EX Version 1.1.1 of the STIX fonts was released for my birthday in 2014. The most recent release is Version 2.10 in December 2020. The last addition was on July 2021, adding the Google fonts.

Installation and usage

If you have a T_EX installation from 2014 or later, then you should already have the STIX fonts installed. To test whether this is the case, try to typeset the following tiny article:

```
\documentclass{article}
\usepackage{stix}
\begin{document}
Some text, and a math formula  $\ccwundercurvearrow$ .
\end{document}
```

This should typeset as

┌

Some text, and a math formula \curvearrowright .

└

To use the STIX fonts, load the STIX package, as in the example. Note that the STIX package has to be loaded *ahead of* the AMS packages. Since the `amsart` document class loads many AMS packages, we have to use the unusual construct:

```

\RequirePackage{stix}
\documentclass{amsart}
\begin{document}
Some text, and a math formula  $\$\ccwundercurvearrow$.$ 
\end{document}

```

This typesets as the previous example.

If your installation is not up-to-date, it is simpler to update it than to try to install the STIX fonts yourself. You can find the STIX fonts at

<https://www.stixfonts.org>

The current version consists of five OpenType font files:

```

STIX2Math.otf
STIX2Text-Regular.otf
STIX2Text-Bold.otf
STIX2Text-Italic.otf
STIX2Text-BoldItalic.otf

```

Follow the installation instructions; they are operating system dependent.

All the math symbols are listed in 19 pages See Figure 8.1, for the second of these 19 pages. These pages show all the symbols and the commands necessary to produce them. So to get



type `\varcarriagereturn`. As in Figure 8.1', a * indicates that there is no bold version of the symbol.

How to find a symbol in the 19 page listing? With perseverance. The symbols are divided into 12 parts; the longest is Relations, about seven pages! Within a part, they are listed by their hexadecimal numbers (really).

8.3 Building new symbols

No matter how many math symbols \LaTeX provides, users always seem to want more. \LaTeX gives you excellent tools to build new symbols from existing ones.

8.3.1 Stacking symbols

In \LaTeX , you can stack symbols above or below other symbols using various commands. Here are a few examples:

To place any symbol above, or below, any other, for instance, \sim , use the `\overset` command. It takes two arguments—the first argument is set in a smaller size above the second argument. The spacing rules of the symbol in the second argument remain valid, i.e., the type remains the same. Since \sim is a binary relation, so is $\overset{\sim}{\sim}$. The `\underset`

\backslash	U+2037	\backtrprime	\Downarrow	U+21E9	\downwhitearrow
\wedge	U+2038	\caretinsert	\Updownarrow	U+21EA	\whitearrowupfrombar
!!	U+203C	\Exclam	\forall	U+2200	\forallforall
-	U+2043	\hyphenbullet*	\complement	U+2201	\complement
??	U+2047	\Question	\exists	U+2203	\existsexists
'''	U+2057	\qprime	\nexists	U+2204	\nexists
○	U+20DD	\enclosecircle	\varnothing	U+2205	\varnothingnothing
□	U+20DE	\enclosesquare*	\emptyset	U+2205	\emptysetset
◇	U+20DF	\enclosediamond*	Δ	U+2206	\increment
△	U+20E4	\enclosetriangle	■	U+220E	\QED*
\mathcal{E}	U+2107	\Eulerconst	∞	U+221E	\inftyfty
\hbar	U+210F	\hbar*	\perp	U+221F	\rightangle
\hslash	U+210F	\hslash	\angle	U+2220	\angle
\Im	U+2111	\Im	\sphericalangle	U+2221	\measuredangle
ℓ	U+2113	\ell	\sphericalangle	U+2222	\sphericalangle
\wp	U+2118	\wp	\therefore	U+2234	\therefore
\Re	U+211C	\Re	\because	U+2235	\because
\mathfrak{U}	U+2127	\mho	\sim	U+223F	\sinewave
turnediota	U+2129	\turnediota	\top	U+22A4	\top
\AA	U+212B	\Angstrom	\bot	U+22A5	\bot
\mathfrak{F}	U+2132	\Finv	\dagger	U+22B9	\hermitmatrix
\aleph	U+2135	\aleph	\measuredrightangle	U+22BE	\measuredrightangle
\beth	U+2136	\beth	\vartriangleleft	U+22BF	\vartrianglelefttriangle
\gimel	U+2137	\gimel	\cdots	U+22EF	\cdots
\daleth	U+2138	\daleth	\oslash	U+2300	\diameter*
\mathcal{G}	U+2141	\Game*	\triangle	U+2302	\house
\mathcal{L}	U+2142	\sansLturned*	\neg	U+2310	\invnot
\mathcal{M}	U+2143	\sansMmirrored*	\sqlozenge	U+2311	\sqlozenge*
\mathcal{Y}	U+2144	\Yup*	\frown	U+2312	\frownline*
\mathcal{P}	U+214A	\PropertyLine*	\smile	U+2313	\smileline*
\Uparrow	U+21A8	\updownarrowbar	\boxplus	U+2317	\boxplusdata*
\lfloor	U+21B4	\linefeed	\neg	U+2319	\turnednot
\lceil	U+21B5	\carriagereturn	\hexagon	U+232C	\varhexagonnlrbonds*
\nwarrow	U+21B8	\barovernorthwestarrow	\triangleright	U+2332	\conictaper*
\barleftarrow	U+21B9	\barleftarrowrightarrowbar	\top	U+2336	\topbot
\circlearrowleft	U+21BA	\acwopencirclearrow	∇	U+2340	\APLnotbackslash*
\circlearrowright	U+21BB	\cwopencirclearrow	\boxup	U+2353	\APLboxupcaret*
\Updownarrow	U+21DE	\nHuparrow*	\boxquestion	U+2370	\APLboxquestion*
\Downarrow	U+21DF	\nHdownarrow*	\rightrightarrows	U+237C	\rightrightarrows*
\dashleftarrow	U+21E0	\leftdasharrow*	\hexagon	U+2394	\hexagon*
\dashrightarrow	U+21E1	\righdasharrow*	\equiv	U+23B6	\bbrktbrk
\downdasharrow	U+21E3	\downdasharrow*	\curvearrowright	U+23CE	\varcarriagereturn*
\leftarrow	U+21E6	\leftwhitearrow	\lrcorner	U+23E0	\obrbrak
\rightarrow	U+21E7	\upwhitearrow	\llcorner	U+23E1	\ubrbrak
\Rightarrow	U+21E8	\rightwhitearrow	\triangle	U+23E2	\trapezium*
			\odot	U+23E3	\benzenr*

Figure 8.1: A sample page from the STIX document


```
\[
  \frac{a}{b} \overset{\normalfont u}{+} \frac{c}{d}
  \overset{\normalfont l}{+} \frac{e}{f}
\]
```

A 2019 addition to L^AT_EX introduces the `\overunderset` command. Here are some examples:

```
\[\overunderset{a}{b}{C}, \overunderset{l}{-}{C}\]
```

typesets as

$$\overset{a}{\underset{b}{C}}, \quad \overset{1}{\underset{-}{C}}.$$

8.3.2 Negating and side-setting symbols

You can *negate* with the `\not` command; for instance, $a \not\in b$ and $a \neq b$ are typed as `$a \not\in b$` and `$a \neq b$`, respectively. It is preferable, however, to use the negated symbols \notin , typed as `\notin`, and \neq , typed as `\neq`. See the negated binary relations table in Appendix A.2. For instance, “ a does not divide b ,” $a \nmid b$, should be typed as `$a \nmid b$`, not as `$a \not\mid b$`, which typesets as $a \not\mid b$. In Section 8.3.3, you learn how to change $a \not\mid b$ to $a \nparallel b$, typed as `$a \mathrel{\not\mid} b$`. However, `$a \nmid b$` is still best.

L^AT_EX provides the `\sideset` command to set symbols at the corners of large operators other than the “corners” (the last four delimiters in Table 7.3). This command takes three arguments:

```
\sideset{ _{ll}^{ul} }{ _{lr}^{ur} }{large_op}
```

where *ll* stands for the symbol to be placed at the lower left, *ul* for upper left, *lr* for lower right, and *ur* for upper right; *large_op* is a large operator. These two examples,

$$\prod_a^c \text{ and } \prod^e$$

are typed as

```
\[
  \sideset{}{_{a}^{c}}{\prod}\text{ and } \sideset{^{e}}{}{\prod}
\]
```

Note that the two first arguments are compulsory, although one or the other may be empty, while the third argument must contain the large operator.

Here is a more meaningful example:

```
\[
  \sideset{}{'}{\sum}_{\substack{i < 10 \\ j < 10}} x_i z_j
\]
```

typeset as

$$\sum'_{\substack{i < 10 \\ j < 10}} x_i z_j$$

In this example, note that prime (') is an automatically superscripted symbol (see Section 7.4.1), so you do not have to type `^'` in the second argument. Typing `\sum'` would not work, since L^AT_EX would place the prime above the sum symbol.

Thus, `\sideset` helps in mixing sub- and superscripts in “limit” positions with others in “nolimit” positions, allowing for a total of six positions in displayed operators with limits. Try

$$\prod_{a_n}^c x_e^i$$

typed as

```
\[
\sideset{_{a}^{c}}{_{e}^{i}}{\prod}_{n}^{r}
\]
```

8.3.3 Changing the type of a symbol

Some symbols are binary relations and some are binary operations (see Section 8.1). In fact, you can force any symbol to behave like either. The `\mathbin` command declares its argument to be a binary operation. For example,

```
\mathbin{\alpha}
```

makes this instance of `\alpha` behave like a binary operation, as in $a \alpha b$, typed as

```
$a \mathbin{\alpha} b$
```

You can use the `\mathrel` command to make a symbol behave like a binary relation, as in the formula $a \alpha b$, typed as

```
$a \mathrel{\alpha} b$
```

You can see

```
 $a \alpha b$  ($a \mathbin{\alpha} b$)
```

```
 $a \alpha b$  ($a \mathrel{\alpha} b$)
```

that a binary relation provides a bit more space than a binary operation. There is an interesting use of `\mathbin` on page 90.

In Section 15.1.6, we discuss the `\DeclareMathOperator` command and its *-ed version, to declare a symbol—or any text or formula—a math operator.

8.4 Math alphabets and symbols

The classification of math symbols in the context of spacing was discussed in Section 8.1. The symbols in a formula can also be classified as *characters from math alphabets* and *math symbols*. In the formula

$$A = \{x \in X \mid x\beta \geq xy > (x+1)^2 - \alpha\}$$

the following characters come from math alphabets:

$$A \quad x \quad X \quad y \quad 1 \quad 2$$

whereas these characters are math symbols:

$$= \quad \{ \quad \in \quad | \quad \beta \quad \geq \quad > \quad (\quad + \quad) \quad - \quad \alpha \quad \}$$

8.4.1 Math alphabets

The letters and digits typed in a math formula come from *math alphabets*. L^AT_EX’s default math alphabet—the one you get if you do not ask for something else—is Computer Modern math italic for *letters*. In the formula $x^2 \vee y_3 = \alpha$, the characters x and y come from this math alphabet. The default math alphabet for *digits* is Computer Modern roman and the digits 2 and 3 in this formula are typeset in Computer Modern roman.

L^AT_EX has a number of commands to switch type style in math. The two most important commands select the bold and italic versions:

Command	Math alphabet	Produces
<code>\mathbf{Greek\ \gamma, \ \Gamma}</code>	math bold	Greek γ, Γ
<code>\mathit{Greek\ \gamma, \ \Gamma}</code>	math italic	<i>Greek γ, Γ</i>

These commands change the style of letters, numbers, and upper case Greek characters. But beware of the pitfalls. For instance, in `\mathit{left-side}` the hyphen typesets as a minus: *left – side*.

There are four more commands that switch math alphabets:

Command	Math Alphabet	Produces
<code>\mathsf{Greek\ \gamma, \ \Gamma}</code>	math sans serif	Greek γ , Γ
<code>\mathrm{Greek\ \gamma, \ \Gamma}</code>	math roman	Greek γ , Γ
<code>\mathtt{Greek\ \gamma, \ \Gamma}</code>	math typewriter	Greek γ , Γ
<code>\mathnormal{Greek\ \gamma, \ \Gamma}</code>	math italic	<i>Greek γ, Γ</i>

Math roman is used in formulas for operator names, such as \sin in $\sin x$, and for text. For operator names, you should use the `\DeclareMathOperator` command or the `*-ed` version, which sets the name of the operator in math roman, and also provides the proper spacing (see Section 8.3.3). For text, you should use the `\text` command (see Section 7.4.6).

The `\mathnormal` command switches to the default math alphabet; it is seldom used in practice.

The Computer Modern fonts include a math bold italic alphabet. To access it, use the `\boldsymbol` command: ***bold italic math***, typed as

```
\boldsymbol{\mathit{bold\ italic\ math}}
```



Rule: Math alphabets vs. text alphabets

Do not use text alphabets in a math formula, except in the argument of a `\text` command.

It may not be easy to see the difference, but some things will not look right or may not align properly.

8.4.2 Math symbol alphabets

You may have noticed that α was not classified as belonging to an alphabet in the example at the beginning of this section. Indeed, α is treated by \LaTeX as a math symbol rather than as a member of a math alphabet. You cannot italicize or slant it, nor is there a sans serif version. For the bold version, use the `\boldsymbol` command to produce it. For instance, α_β , is typed as

```
\boldsymbol{\alpha}_{\boldsymbol{\beta}}
```

Note that β appears in a small size in α_β .

Four “alphabets of symbols” are built into \LaTeX .

Greek The examples α, β, Γ are typed as

```
\alpha, \beta, \Gamma
```

See Section A.1 for the symbol tables.

Calligraphic an uppercase-only alphabet invoked with the `\mathcal` command. The examples $\mathcal{A}, \mathcal{C}, \mathcal{E}$ are typed as

```
\mathcal{A}, \mathcal{C}, \mathcal{E}
```


Euler Fraktur invoked by the `\mathfrak` command. The examples \mathfrak{n} , \mathfrak{N} , \mathfrak{p} , \mathfrak{P} are typed as

```
\mathfrak{n}, \mathfrak{N}, \mathfrak{p}, \mathfrak{P}
```

Blackboard bold uppercase-only math alphabet, invoked with `\mathbb`. The examples \mathbb{A} , \mathbb{B} , \mathbb{C} are typed as

```
\mathbb{A}, \mathbb{B}, \mathbb{C}
```

8.4.3 Bold math symbols

In math, most characteristics of a font are specified by L^AT_EX. One exception is bold-face. To make a *letter* bold from a math alphabet within a formula, use the `\mathbf` command. For instance, in

we choose the vector \mathbf{v}

the bold \mathbf{v} is produced by `\mathbf{v}`.

To obtain bold math *symbols*, use the `\boldsymbol` command. For example, the bold symbols

$$5 \quad \alpha \quad \Lambda \quad \mathcal{A} \quad \rightarrow \quad A$$

are typed as

```
\[
\boldsymbol{5} \quad \quad \quad \boldsymbol{\alpha}
\quad \boldsymbol{\Lambda} \quad \boldsymbol{\mathcal{A}}
\quad \boldsymbol{\rightarrow} \quad \boldsymbol{A}
\]
```

Note that `\boldsymbol{A}` typesets as \mathbf{A} , a bold math italic A. To get an upright \mathbf{A} , type `\mathbf{A}`. The digit 5 did not really need `\boldsymbol`; `\mathbf{5}` gives the same result.

To make an entire formula bold, use the `\mathversion{bold}` command, as in

```
\mathversion{bold} $a \equiv c \pmod{\theta}$
```

which typesets as $\mathbf{a} \equiv \mathbf{c} \pmod{\theta}$. Note that the `\mathversion{bold}` command is given *before the formula*.

To typeset **AMS**, type

```
\boldsymbol{\mathcal{A}} \boldsymbol{\mathcal{M}}
\boldsymbol{\mathcal{S}}
```

or

```
\boldsymbol{ \mathcal{AMS} }
```

or

```
{\mathversion{bold} $ \mathcal{AMS} $}
```

Within the scope of `\mathversion{bold}`, you can undo its effect with

```
\mathversion{normal}
```

Not all symbols have bold variants. For example, if you type

```
$\sum \quad \boldsymbol{\sum}$
```

you get $\sum \quad \sum$, two identical symbols. If you want to obtain a bold version, use the *poor man's bold* invoked by the `\pmb` command. This command typesets the symbol three times very close to one another producing a bold symbol of some quality. Note that `\pmb` does destroy the type of the symbol, `\pmb{\sum}` is no longer spaced like a large operator. To make it into a large operator, declare in the preamble

```
\DeclareMathOperator{\boldsum}{\pmb{\sum}}
```

and

```
\DeclareMathOperator*{\boldsumlim}{\pmb{\sum}}
```

Compare the following four variants of sum:

$$\sum_{i=1}^n i^2 \quad \sum_{i=1}^n i^2 \quad \sum_{i=1}^n i^2 \quad \sum_{i=1}^n i^2$$

The first sum is typed (in displayed math mode) as

```
\sum_{i = 1}^n i^2
```

The second uses poor man's bold, but does not declare the result to be a large operator:

```
\pmb{\sum}_{i = 1}^n i^2
```

The third uses the math operator declared:

```
\boldsum_{i = 1}^n i^2
```

The fourth uses the math operator with limit declared:

```
\boldsumlim_{i = 1}^n i^2
```

The `\bm` package provides the `\bm`, which makes everything bold in its argument. So `\bm{f(x) = ax^2 + bx + c = 0}` makes the whole formula

8.4.4 Size changes

There are four math font sizes, invoked by the command declarations:

- `\displaystyle`, normal size for displayed formulas;
- `\textstyle`, normal size for inline formulas;
- `\scriptstyle`, normal size for subscripted and superscripted symbols;
- `\scriptscriptstyle`, normal size for doubly subscripted and superscripted symbols.

These commands control a number of style parameters in addition to the size. Compare the two fractions

$$\frac{1}{2 + \frac{1}{3}} \quad \frac{1}{2 + \frac{1}{3}}$$

typed as

```
\[
  \frac{1}{\displaystyle 2 + \frac{1}{3}} \quad
  \frac{1}{2 + \frac{1}{3}}
\]
```

8.4.5 Continued fractions

In addition to the `\frac`, `\dfrac`, and `\tfrac` commands (see Section 7.4.1), \LaTeX makes typesetting continued fractions even easier by providing the `\cfrac` command. The `\cfrac` command takes an optional argument, `l` or `r`, to place the numerator on the left or on the right. For example,

$$2 + \frac{1}{3 + \cdots} \quad 2 + \frac{1}{3 + \cdots}$$

is typed as

```
\[
  \cfrac{1}{2 + \cfrac{1}{3 + \cdots}} \quad
  \cfrac[l]{1}{2 + \cfrac[l]{1}{3 + \cdots}}
\]
```

8.5 Vertical spacing

As a rule, all horizontal and vertical spacing in a math formula is done by \LaTeX . Nevertheless, you often need to adjust horizontal spacing (see Section 8.1). There is seldom a need to adjust vertical spacing, but there are a few exceptions.

The formula $\sqrt{a} + \sqrt{b}$ does not look quite right, because the square roots are not uniform. You can correct this with `\mathstrut` commands, which inserts an invisible vertical space:

```
$\sqrt{\mathstrut a} + \sqrt{\mathstrut b}$
```

typesets as $\sqrt{a} + \sqrt{b}$. See Section 3.8.5 for struts in general.

Another way to handle this situation is with the `\vphantom` (vertical phantom) command, which measures the height of its argument and places a math strut of that height into the formula. So

```
$\sqrt{\vphantom{b} a} + \sqrt{b}$
```

also prints uniform square roots, $\sqrt{a} + \sqrt{b}$. The `\vphantom` method is more versatile than the previous one.

Here is a more complicated example from a research article:

$$\Theta_i = \bigcup (\Theta(\overline{a \wedge b}, \overline{a} \wedge \overline{b}) \mid a, b \in B_i) \vee \bigcup (\Theta(\overline{a \vee b}, \overline{a} \vee \overline{b}) \mid a, b \in B_i),$$

typed as

```
\[
\Theta_i = \bigcup \big( \Theta(\overline{a \wedge b},
\overline{\vphantom{b}a} \wedge \overline{b})
\mid a, \ b \in B_i \big)
\vee \bigcup \big( \Theta(\overline{a \vee b},
\overline{\vphantom{b}a} \vee \overline{b} )
\mid a, \ b \in B_i \big),
\]
```

Another useful command for vertical spacing is the `\smash` command. It directs \LaTeX to pretend that its argument does not protrude above or below the line in which it is typeset.

For instance, the two lines of this admonition:

It is *very important* that you memorize the integral $\frac{1}{\int f(x) dx} = 2g(x) + C$, which will appear on the next test.

are too far apart because \LaTeX had to make room for the fraction. However, in this instance, the extra vertical space is not necessary because the second line is very short. To correct this, place the formula in the argument of a `\smash` command:

It is `\emph{very important}` that you memorize the
integral `$\smash{\frac{1}{\int f(x) \, dx}} = 2g(x) + C$`,
which will appear on the next test.

L^AT_EX produces the following:

┌
It is *very important* that you memorize the integral $\frac{1}{\int f(x) dx} = 2g(x) + C$,
which will appear on the next test.
└

An optional argument to the `\smash` command controls which part of the formula is ignored, `t` to smash the top and `b` to smash the bottom.

8.6 Tagging and grouping

You can attach a name to an equation using the `\tag` command. In the `equation` or `equation*` environments,

`\tag{name}`

attaches the tag *name* to the equation—*name* is typeset as text. The tag replaces the number for an equation.

Recall that the numbering of an equation is *relative*, that is, the number assigned to an equation is relative to the placement of the equation with respect to other equations in the document. An equation tag, on the other hand, is *absolute*—the tag remains the same even if the equation is moved.

If there is a tag, the `equation` and the `equation*` environments are equivalent. For example,

$$(Int) \qquad \int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

may be typed as

```
\begin{equation*}
  \int_{-\infty}^{\infty} e^{-x^2} \, dx
  = \sqrt{\pi} \tag{Int}
\end{equation*}
```

or

```
\begin{equation}
  \int_{-\infty}^{\infty} e^{-x^2} \, dx
  = \sqrt{\pi} \tag{Int}
\end{equation}
```

or

```
\[
  \int_{-\infty}^{\infty} e^{-x^2} \, dx
  = \sqrt{\pi} \tag{Int}
\]
```

The `\tag*` command is the same as `\tag` except that it does not automatically enclose the tag in parentheses. To get

$$A-B \qquad \int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

type

```
\begin{equation}
  \int_{-\infty}^{\infty} e^{-x^2} \, dx = \sqrt{\pi}
  \tag*{A--B}
\end{equation}
```

Tagging allows numbered variants of equations. For instance, the equation

$$(1) \qquad A^{[2]} \diamond B^{[2]} \cong (A \diamond B)^{[2]}$$

may need a variant:

$$(1') \qquad A^{\langle 2 \rangle} \diamond B^{\langle 2 \rangle} \equiv (A \diamond B)^{\langle 2 \rangle}$$

If the label of the first equation is `E:first`, then the second equation may be typed as follows:

```
\begin{equation}\tag{\ref{E:first}}{'}{'}
  A^{\langle 2 \rangle} \diamond B^{\langle 2 \rangle}
  \equiv (A \diamond B)^{\langle 2 \rangle}
\end{equation}
```

Such a tag is absolute in the sense that it does not change if the equation is moved. But if it references a label and the number generated by \LaTeX for the label changes, the tag changes.

In contrast, *grouping* applies to a group of *adjacent* equations. Suppose the last equation was numbered (1) and the next group of equations is to be referred to as (2), with individual equations numbered as (2a), (2b), and so on. Enclosing these equations in a `subequations` environment accomplishes this goal. For instance,

$$(1a) \qquad A^{[2]} \diamond B^{[2]} \cong (A \diamond B)^{[2]}$$

and its variant

$$(1b) \qquad A^{\langle 2 \rangle} \diamond B^{\langle 2 \rangle} \equiv (A \diamond B)^{\langle 2 \rangle}$$

are typed as

```
\begin{subequations}\label{E:joint}
  \begin{equation}\label{E:original}
    A^{[2]} \diamond B^{[2]} \cong (A \diamond B)^{[2]}
  \end{equation}
\end{subequations}
```

```

\begin{equation}\label{E:modified}
  A^{\{\backslashlangle 2 \rangle\}} \diamond B^{\{\backslashlangle 2\rangle\}}
  \equiv (A \diamond B)^{\{\backslashlangle 2\rangle\}}
\end{equation}
\end{subequations}

```

Referring to these equations, you find that:

- `\eqref{E:joint}` resolves to (1),
- `\eqref{E:original}` resolves to (1a),
- `\eqref{E:modified}` resolves to (1b).

Note that in this example, references to the second and third labels produce numbers, (1a) and (1b), that also appear in the typeset version. The group label, E: joint, references the entire group, but (1) does not appear in the typeset version unless referenced.

A `subequations` environment can contain the multiline math constructs discussed in Chapter 9 (see Section 9.4.4).

8.7 Miscellaneous

Generalized fractions

The generalized fraction command provides the facility to typeset many variants of fractions and binomials, such as $\frac{a+b}{c}$ and $\left] \frac{a+b}{c} \right[$. The syntax is

```

\genfrac{left-delim}{right-delim}{thickness}{mathstyle}
{numerator}{denominator}

```

where:

- *left-delim* is the left delimiter for the formula (default: none);
- *right-delim* is the right delimiter for the formula (default: none);
- *thickness* is the thickness of the fraction line, in the form *x*pt (default: the normal weight, 0.4pt), for instance, 12pt for 12 point thickness;
- *mathstyle* is one of:
 - 0 for `\displaystyle`;
 - 1 for `\textstyle`;
 - 2 for `\scriptstyle`;
 - 3 for `\scriptscriptstyle`;

- Default: Depends on the context. If the formula is being set in display style, then the default is 0, and so on;

- *numerator* is the numerator;
- *denominator* is the denominator.

All arguments must be specified. The empty argument, {}, gives the default value.

Here are some illustrations of generalized fractions.

Example 1

`\frac{numerator}{denominator}`

is the same as

`\genfrac{}{}{}{}{numerator}{denominator}`

Example 2

`\dfrac{numerator}{denominator}`

is the same as

`\genfrac{}{}{}{0}{numerator}{denominator}`

Example 3

`\tfrac{numerator}{denominator}`

is the same as

`\genfrac{}{}{}{1}{numerator}{denominator}`

Example 4

`\binom{numerator}{denominator}`

is the same as

`\genfrac{({)}{0pt}{}{numerator}{denominator}`

Example 5 Here are some more examples:

$$\frac{a+b}{c} \quad \frac{a+b}{c} \quad \frac{a+b}{c} \quad \frac{a+b}{c} \quad \left[\begin{array}{c} a+b \\ c \end{array} \right] \quad \left] \begin{array}{c} a+b \\ c \end{array} \right[$$

typed as


```
\[
\frac{a + b}{c} \quad
\genfrac{}{}{1pt}{}{a + b}{c} \quad
\genfrac{}{}{1.5pt}{}{a + b}{c} \quad
\genfrac{}{}{2pt}{}{a + b}{c} \quad
\genfrac{[]{}{0pt}{}{a + b}{c} \quad
\genfrac{}{}{[]}{0pt}{}{a + b}{c}
\]
```

$$\frac{a + b}{c} \quad \frac{a + b}{c}$$

typed as

```
\[
\frac{a + b}{c} \quad
\genfrac{}{}{0.4pt}{}{a + b}{c}
\]
```

You can choose the delimiters from Table 7.3.

If a `\genfrac` construct is used repeatedly, you should name it. See Section 15.1 for custom commands.

Boxed formulas

The `\boxed` command puts its argument in a box, as in

(2)
$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

typed as

```
\begin{equation}
\boxed{\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}}
\end{equation}
```

The `\boxed` command can also be used in the argument of a `\text` command. Note that

```
\fbox{Hello world}
```

and

```
\boxed{\text{Hello world}}
```

produce the same Hello world.

Morten Høgholm's `mathtools` package contains many variants of boxes.



Multiline math displays

9.1 Visual Guide

\LaTeX is about typesetting text and math. It knows a lot about typesetting inline formulas, but not much about how to display a multiline formula to best reflect its meaning in a visually pleasing way. So you have to decide the visual structure of a multiline formula and then use the tools provided by \LaTeX to code and typeset it.

For many mathematical documents the three constructs of Chapter 1 suffice: *simple* and *annotated* alignments, and the *cases* construct. To help you choose the appropriate tool for more complicated constructs, we start by introducing the basic concepts and constructions with the *Visual Guide* of Figure 9.1.

9.1.1 Columns

Multiline math formulas are displayed in *columns*. The columns are either *adjusted*, that is, centered, or set flush left or right, or *aligned*, that is, an alignment point is designated for each column and for each line. Moreover, the columns are either separated by the *intercolumn space* or adjacent with no separation.

Adjusted environments

$$\begin{array}{cc}
 \begin{array}{c}
 x_1x_2 + x_1^2x_2^2 + x_3 \\
 x_1x_3 + x_1^2x_3^2 + x_2 \\
 x_1x_2x_3
 \end{array}
 &
 \begin{array}{c}
 (x_1x_2x_3x_4x_5x_6)^2 \\
 + (x_1x_2x_3x_4x_5 + x_1x_3x_4x_5x_6 + x_1x_2x_4x_5x_6 + x_1x_2x_3x_5x_6)^2 \\
 + (x_1x_2x_3x_4 + x_1x_2x_3x_5 + x_1x_2x_4x_5 + x_1x_3x_4x_5)^2
 \end{array}
 \end{array}$$

gather
one column, centered
multline
flush left, centered, flush right

Adjusted subsidiary environments

$$\begin{array}{ccc}
 \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}
 &
 \begin{array}{ccccc}
 a+b+c & uv & x-y & 27 \\
 a+b & u+v & z & 134
 \end{array}
 &
 f(x) = \begin{cases} -x^2, & \text{if } x < 0; \\ \alpha + x, & \text{if } 0 \leq x \leq 1; \\ x^2, & \text{otherwise.} \end{cases}
 \end{array}$$

matrix
multicolumn, centered
array
multicolumn
each column adjusted independently
cases
columns flush left

Aligned environments

$$\begin{array}{cccc}
 f(x) = x + yz & g(x) = x + y + z & f(x) = x + yz & g(x) = x + y + z \\
 h(x) = xy + xz + yz & k(x) = (x + y)(x + z)(y + z) & h(x) = xy + xz + yz & k(x) = (x + y)(x + z)(y + z)
 \end{array}$$

align
multicolumn, aligned
flalign
multicolumn, aligned

$$\begin{array}{rcl}
 (17) & a_{11}x_1 + a_{12}x_2 + a_{13}x_3 & = y_1 \\
 (18) & a_{21}x_1 + a_{22}x_2 & + a_{24}x_4 = y_2 \\
 (19) & a_{31}x_1 & + a_{33}x_3 + a_{34}x_4 = y_3
 \end{array}$$

alignat
multicolumn, aligned

Aligned subsidiary environment

$$\begin{array}{c}
 (3.4) \quad 0 = \langle \dots, 0, \dots, \overset{i}{d}, \dots, 0, \dots \rangle \wedge \langle \dots, 0, \dots, \overset{j}{d}, \dots, 0, \dots \rangle \\
 \equiv \langle \dots, 0, \dots, \overset{j}{d}, \dots, 0, \dots \rangle \pmod{\Theta}
 \end{array}$$

split
one column, aligned

Figure 9.1: The *Visual Guide* for multiline math formulas.

One column

As in Chapter 1, we start with a simple align:

$$\begin{aligned} r^2 &= s^2 + t^2, \\ 2u + 1 &= v + w^\alpha \end{aligned}$$

This is a single column, aligned at the = signs, and coded with the `align` environment (see Section 1.7.3).

Two columns

The annotated align, coded with the `align` environment (see Section 1.7.3),

$$\begin{aligned} x &= x \wedge (y \vee z) && \text{(by distributivity)} \\ &= (x \wedge y) \vee (x \wedge z) && \text{(by condition (M))} \\ &= y \vee z \end{aligned}$$

has two columns. The first column is aligned like our example of simple align, but the second column is aligned flush left. There is a sizeable intercolumn space.

9.1.2 Subsidiary math environments

The cases example in Section 1.7.4:

$$f(x) = \begin{cases} -x^2, & \text{if } x < 0; \\ \alpha + x, & \text{if } 0 \leq x \leq 1; \\ x^2, & \text{otherwise.} \end{cases}$$

introduces a new concept. The part of the formula to the right of = is a multiline construct. This is an example of a *subsidiary math environment* that can only be used *inside another math environment*. It creates a “large math symbol,” in this case

$$\begin{cases} -x^2, & \text{if } x < 0; \\ \alpha + x, & \text{if } 0 \leq x \leq 1; \\ x^2, & \text{otherwise.} \end{cases}$$

So the cases example:

$$f(x) = \text{large math symbol}$$

is a single line displayed formula, where “large math symbol” is replaced by the cases construct.

9.1.3 Adjusted columns

An *adjusted column* is either set *centered*, or *flush left*, or *flush right*. This may happen by default, built into the environment, or be so specified in the code. For instance, in the displayed formula

$$\begin{array}{l} x_1x_2 + x_1^2x_2^2 + x_3, \\ x_1x_3 + x_1^2x_3^2 + x_2 \end{array}$$

typeset with the `gather` environment, by default all the lines are centered.

On the other hand, in

$$\left(\begin{array}{ccc} 1 & 100 & 115 \\ 201 & 0 & 1 \end{array} \right)$$

coded with the `array` subsidiary math environment, the first column is flush left, the second centered, the third flush right.

9.1.4 Aligned columns

Aligned columns, on the other hand, are only of one kind, aligned by you. For instance,

$$\begin{array}{ll} f(x) = x + yz & g(x) = x + y + z \\ h(x) = xy + xz + yz & k(x) = (x + y)(x + z)(y + z) \end{array}$$

is coded with the `alignat` environment. It has two aligned columns, both aligned at the = signs.

9.1.5 Touring the Visual Guide

Figure 9.1, the *Visual Guide*, shows thumbnail pictures of the various kinds of multiline math environments and subsidiary math environments.

The first part of the *Visual Guide* illustrates `gather` and `multline`. The `gather` environment is a one-column, centered math environment—discussed in Section 9.2—which is used to display a *number of formulas* collected into one multiline formula. In contrast, `multline`—discussed in Section 9.3—displays *one long formula* in a number of lines. The first line is set flush left, the last line set flush right, and the rest (if any) of the lines are centered.

The third part of the *Visual Guide* illustrates the `align` environment and two of its variants, `alignat` and `flalign`, discussed in Section 9.5.

Three adjusted subsidiary math environments—`matrix`, `cases`, and `array`—are illustrated in second part of the *Visual Guide* and presented in Section 9.7.

The aligned subsidiary math environments `aligned` and `gathered` look just like the `align` and `gather` environments, so they are not illustrated in the *Visual Guide*. The `aligned` and `gathered` environments—along with `\itemref`—are discussed in Section 9.6, along with the `split` subsidiary math environment; this last one is illustrated in the last part of the *Visual Guide*.

9.2 Gathering formulas

The `gather` environment groups a number of one-line formulas, each centered on a separate line:

$$\begin{aligned} (1) \quad & x_1x_2 + x_1^2x_2^2 + x_3, \\ (2) \quad & x_1x_3 + x_1^2x_3^2 + x_2, \\ (3) \quad & x_1x_2x_3. \end{aligned}$$

Formulas (1)–(3) are typed as follows:

```
\begin{gather}
x_{1} x_{2}+x_{1}^{\wedge}2} x_{2}^{\wedge}2} + x_{3},\label{E:1.1}\\
x_{1} x_{3}+x_{1}^{\wedge}2} x_{3}^{\wedge}2} + x_{2},\label{E:1.2}\\
x_{1} x_{2} x_{3}.\label{E:1.3}
\end{gather}
```



Rule: gather environment

1. Lines are separated with `\\`. Do not type `\\` at the end of the last line!
2. Each line is numbered unless it has a `\tag` or `\notag` on the line before the line separator `\\`.
3. No blank lines are permitted within the environment.

The `gather*` environment is like `gather`, except that all lines are unnumbered. They can still be `\tag`-ged.

It would seem natural to code formulas (1)–(3) with three `equation` environments:

```
\begin{equation}
x_{1} x_{2}+x_{1}^{\wedge}2} x_{2}^{\wedge}2} + x_{3},\label{E:1.1}
\end{equation}
\begin{equation}
x_{1} x_{3}+x_{1}^{\wedge}2} x_{3}^{\wedge}2} + x_{2},\label{E:1.2}
\end{equation}
\begin{equation}
x_{1} x_{2} x_{3}.\label{E:1.3}
\end{equation}
```

Note how bad this looks typeset:

$$(1) \quad x_1x_2 + x_1^2x_2^2 + x_3,$$

$$(2) \quad x_1x_3 + x_1^2x_3^2 + x_2,$$

$$(3) \quad x_1x_2x_3.$$

9.3 Splitting long formulas

The `multline` environment is used to split one very long formula into several lines. The first line is set flush left, the last line is set flush right, and the middle lines are centered:

$$(4) \quad (x_1x_2x_3x_4x_5x_6)^2 \\ + (y_1y_2y_3y_4y_5 + y_1y_3y_4y_5y_6 + y_1y_2y_4y_5y_6 + y_1y_2y_3y_5y_6)^2 \\ + (z_1z_2z_3z_4z_5 + z_1z_3z_4z_5z_6 + z_1z_2z_4z_5z_6 + z_1z_2z_3z_5z_6)^2 \\ + (u_1u_2u_3u_4 + u_1u_2u_3u_5 + u_1u_2u_4u_5 + u_1u_3u_4u_5)^2$$

This formula is typed as

```
\begin{multline}\label{E:mm2}
(x_{1} x_{2} x_{3} x_{4} x_{5} x_{6})^2\\
+ (y_{1} y_{2} y_{3} y_{4} y_{5}
+ y_{1} y_{3} y_{4} y_{5} y_{6}
+ y_{1} y_{2} y_{4} y_{5} y_{6}
+ y_{1} y_{2} y_{3} y_{5} y_{6})^2\\
+ (z_{1} z_{2} z_{3} z_{4} z_{5}
+ z_{1} z_{3} z_{4} z_{5} z_{6}
+ z_{1} z_{2} z_{4} z_{5} z_{6}
+ z_{1} z_{2} z_{3} z_{5} z_{6})^2\\
+ (u_{1} u_{2} u_{3} u_{4} + u_{1} u_{2} u_{3} u_{5}
+ u_{1} u_{2} u_{4} u_{5} + u_{1} u_{3} u_{4} u_{5})^2
\end{multline}
```



Rule: `multline` environment

1. Lines are separated with `\\`. Do not type `\\` at the end of the last line!

2. The formula is numbered *as a whole* unless it is `\tag-`ged or the numbering is suppressed with `\notag`. (Alternatively, use the `multline*` environment.)
3. No blank lines are permitted within the environment.
4. Each line is a subformula (see Section 9.4.2).

If you are very observant, you may have noticed that we failed to type `{}`+ following the line separators of the formula. In Section 8.1.2, you were told that this omission would result in the second line being typeset as

$$+(y_1y_2y_3y_4y_5 + y_1y_3y_4y_5y_6 + y_1y_2y_4y_5y_6 + y_1y_2y_3y_5y_6)^2$$

The `multline` environment, however, knows that a long formula is being broken and so typesets `+` as a binary operation.



Tip Do not write `multiline` for `multline`; it results in the message:

`! LaTeX Error: Environment multiline undefined.`

In the `multline*` environment, the formula is not numbered but can be `\tag-`ged.

The indentation of the first and last lines is controlled by the `\multlinegap` length command, with a default of 10 points, unless there is a tag on one of those lines. You can adjust the indentation by enclosing the `multline` environment in a `setlength` environment (see Section 15.5.2), as follows:

```
\begin{multline*}
(x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}})^{\{2\}}\backslash
+ (x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}}
+ x_{\{1\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}}
+ x_{\{1\}} x_{\{2\}} x_{\{4\}} x_{\{5\}} x_{\{6\}}
+ x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{5\}} x_{\{6\}})^{\{2\}}\backslash
+ (x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} + x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{5\}}
+ x_{\{1\}} x_{\{2\}} x_{\{4\}} x_{\{5\}} + x_{\{1\}} x_{\{3\}} x_{\{4\}})^{\{2\}}
\end{multline*}
\begin{setlength}{\multlinegap}{0pt}
\begin{multline*}
(x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}})^{\{2\}}\backslash
+ (x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}}
+ x_{\{1\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}}
+ x_{\{1\}} x_{\{2\}} x_{\{4\}} x_{\{5\}} x_{\{6\}}
+ x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{5\}} x_{\{6\}})^{\{2\}}\backslash
```



```

+ (x_{1} x_{2} x_{3} x_{4} + x_{1} x_{2} x_{3} x_{5}
+ x_{1} x_{2} x_{4} x_{5} + x_{1} x_{3} x_{4})^2}
\end{multline*}
\end{setlength}

```

which typesets as

$$\begin{aligned}
 & (x_1 x_2 x_3 x_4 x_5 x_6)^2 \\
 & + (x_1 x_2 x_3 x_4 x_5 + x_1 x_3 x_4 x_5 x_6 + x_1 x_2 x_4 x_5 x_6 + x_1 x_2 x_3 x_5 x_6)^2 \\
 & + (x_1 x_2 x_3 x_4 + x_1 x_2 x_3 x_5 + x_1 x_2 x_4 x_5 + x_1 x_3 x_4)^2
 \end{aligned}$$

$$\begin{aligned}
 & (x_1 x_2 x_3 x_4 x_5 x_6)^2 \\
 & + (x_1 x_2 x_3 x_4 x_5 + x_1 x_3 x_4 x_5 x_6 + x_1 x_2 x_4 x_5 x_6 + x_1 x_2 x_3 x_5 x_6)^2 \\
 & + (x_1 x_2 x_3 x_4 + x_1 x_2 x_3 x_5 + x_1 x_2 x_4 x_5 + x_1 x_3 x_4)^2
 \end{aligned}$$

Notice that the second variant is not indented.

Any line of a `multline` environment can be typeset flush left or right by making it the argument of a `\shoveleft` or `\shoveright` command, respectively (same with `multline*`). For instance, to typeset the second line of formula (4) flush left, as in

$$\begin{aligned}
 & (x_1 x_2 x_3 x_4 x_5 x_6)^2 \\
 & + (x_1 x_2 x_3 x_4 x_5 + x_1 x_3 x_4 x_5 x_6 + x_1 x_2 x_4 x_5 x_6 + x_1 x_2 x_3 x_5 x_6)^2 \\
 & + (x_1 x_2 x_3 x_4 + x_1 x_2 x_3 x_5 + x_1 x_2 x_4 x_5 + x_1 x_3 x_4 x_5)^2
 \end{aligned}$$

type the formula as follows:

```

\begin{multline*}
(x_{1} x_{2} x_{3} x_{4} x_{5} x_{6})^2\\
\shoveleft{+ (x_{1} x_{2} x_{3} x_{4} x_{5}
+ x_{1} x_{3} x_{4} x_{5} x_{6}
+ x_{1} x_{2} x_{4} x_{5} x_{6}
+ x_{1} x_{2} x_{3} x_{5} x_{6})^2}}
+ (x_{1} x_{2} x_{3} x_{4} + x_{1} x_{2} x_{3} x_{5}
+ x_{1} x_{2} x_{4} x_{5} + x_{1} x_{3} x_{4} x_{5})^2}
\end{multline*}

```

Observe that the entire line is the argument of the `\shoveleft` command, which is followed by `\\` unless it is the last line of the environment.

9.4 Some general rules

9.4.1 General rules

Even though you have only seen a few examples of multiline math environments, I venture to point out now that the multiline math environments and subsidiary math environments share a number of rules.



Rule: Multiline math environments

1. Lines are separated with `\\`. Do not type `\\` at the end of the last line!
2. No blank lines are permitted within an environment.
3. No blank line before the environment.
4. If an environment contains more than one formula, then, as a rule, each formula is numbered separately. If you add a `\label` command to a line, then the equation number generated for that line can be cross-referenced.
5. You can suppress the numbering of a line by using a `\notag` command on the line.
6. You can also override numbering with the `\tag` command, which works just as it does for equations (see Section 8.6).
7. `\tag` and `\label` should always precede the line separator `\\` for lines that are regarded as formulas in their own right. For instance, the lines of the `multline` environment cannot be individually numbered or tagged. The `\tag` command works for individual lines, not for the environment as a whole.
8. For cross-referencing, use `\label`, `\ref`, and `\eqref` in the same way you would for an equation (see Section 7.3).
9. Each multiline math environment has a `*-ed` form, which suppresses numbering. Individual formulas can still be `\tag`-ged.

A `\notag` command placed after the environment is ignored, but a `\tag` command gives the message

```
! Package amsmath Error: \tag not allowed here.
```

9.4.2 Subformula rules

A formula in the `multline` environment is split into a number of parts by `\\` commands; for instance, formula (4) is split into three parts:

1. $(x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}})^{\{2\}}$
2. $+ (x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} + x_{\{1\}} x_{\{3\}} x_{\{4\}} x_{\{5\}} x_{\{6\}} + x_{\{1\}} x_{\{2\}} x_{\{4\}} x_{\{5\}} x_{\{6\}} + x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{5\}} x_{\{6\}})^{\{2\}}$
3. $+ (x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{4\}} + x_{\{1\}} x_{\{2\}} x_{\{3\}} x_{\{5\}} + x_{\{1\}} x_{\{2\}} x_{\{4\}} x_{\{5\}} + x_{\{1\}} x_{\{3\}} x_{\{4\}} x_{\{5\}})^{\{2\}}$

Such parts of a formula are called *subformulas*.

The first line of the aligned formula $r^2 = s^2 + t^2$ —from the simple alignment example in Section 1.7.3—which is typed as

$$r^{\{2\}} \&= s^{\{2\}} + t^{\{2\}}$$

is split into two parts:

1. $r^{\{2\}}$
2. $= s^{\{2\}} + t^{\{2\}}$

In general, in a line of an aligned formula, the first part is everything between the beginning of the line and the first $\&$ symbol. There can then be a number of parts delimited by two consecutive $\&$ symbols. Finally, the last part is from the last $\&$ symbol to the end of the line or the line separator $\backslash\backslash$. These parts are also called *subformulas*.

Here are the last of the general rules.



Rule: Subformula

1. Each subformula must be a formula that L^AT_EX can typeset independently.
2. If a subformula starts with the binary operation $+$ or $-$, type it as $\{+\}$ or $\{-\}$.
3. If a subformula ends with the binary operation $+$ or $-$, type it as $\{+\}$ or $\{-\}$.

Suppose that you want to split the formula

$$x_1 + y_1 + \left(\sum_{i < 5} \binom{5}{i} + a^2 \right)^2$$

just before the binomial coefficient. Try

```
\begin{multline}
  x_{\{1\}} + y_{\{1\}} + \left( \sum_{\{i < 5\}} \backslash\backslash
    \binom{\{5\}}{\{i\}} + a^{\{2\}} \right)^{\{2\}}
\end{multline}
```

When typesetting this formula, you get the message

```
! Missing \right. inserted.
```

because the first subformula violates the first subformula rule.

```
x_{1} + y_{1} + \left( \sum_{i < 5}
```

cannot be typeset by \LaTeX because the `\left(` command must be matched by the `\right` command and some delimiter.

Testing for the first subformula rule is easy. Split the formula into its subformulas, and test each subformula separately by typesetting it.

9.4.3 Breaking and aligning formulas

You do not have to know where and how to break inline math formulas because \LaTeX does all the work for you.

Unfortunately, multiline formulas are different. \LaTeX gives you excellent tools for displaying multiline math formulas, but offers you no advice on deciding where to break a long formula into lines. And that is how it should be. You, the author, are the only judge of where to break a long formula so that the result is mathematically informative and follows the traditions of mathematical typesetting.

A strict set of rules is formulated in *Mathematics into Type* by Ellen Swanson, Arlene Ann O'Sean, and Antoinette Tingley Schleyer [42]. I state only three.



Rule: Breaking displayed formulas

1. Try to break a long formula *before* a binary relation or binary operation.
2. If you break a formula before a `+` or `-`, start the next line with `{}``+` or `{}``-`.
3. If you break a formula within a bracket, indent the next line so that it begins *to the right of* the opening bracket.

Formula (4) on page 216 illustrates the first rule. Here is an illustration of the third rule:

$$f(x, y, z, u) = [(x + y + z) \times (x^2 + y^2 + z^2 - 1) \\ \times (x^3 + y^3 + z^3 - u) \times (x^4 + y^4 + z^4 + u)]^2$$

The rules for aligning columns are similar.

**Rule: Aligning columns**

1. Try to align columns at a binary relation or a binary operation.
2. If you align a column at a binary relation, put the `&` symbol immediately *to the left* of the binary relation.
3. If you align a column at the binary operation `+` or `-`, put the `&` symbol to the left of the binary operation.

9.4.4 Numbering groups of formulas

With most constructs in this chapter, you have a number of equations typeset together, arranged in some way, aligned or adjusted. Each equation is numbered separately, unless `\tagged` or `\notagged`. Often, you may want the equations to share a common number, but still be able to reference each equation separately.

You can change the numbering of the equations on page 215 in formulas (1)–(3) to (1), (1a), and (1b) as follows:

```
\begin{gather}
  x_{1} x_{2} + x_{1}^{2} x_{2}^{2} + x_{3}, \\
  \label{E:1} \\
  x_{1} x_{3} + x_{1}^{2} x_{3}^{2} + x_{2}, \\
  \tag{\ref{E:1}a} \\
  x_{1} x_{2} x_{3}; \tag{\ref{E:1}b}
\end{gather}
```

produces the desired result:

$$\begin{array}{ll}
 (1) & x_1 x_2 + x_1^2 x_2^2 + x_3, \\
 (1a) & x_1 x_3 + x_1^2 x_3^2 + x_2, \\
 (1b) & x_1 x_2 x_3;
 \end{array}$$

To obtain (1') or (1') type

```
\tag{\ref{E:1}$'$}
```

or

```
\tag{(\ref{E:1}\textquoteright)}
```

and for (1_a), type

```
\tag{\ref{E:1}$\}_{\text{a}}$}
```

Alternatively, you may include the `gather` environment in a `subequations` environment (see Section 8.6):

$$(5a) \quad x_1x_2 + x_1^2x_2^2 + x_3,$$

$$(5b) \quad x_1x_3 + x_1^2x_3^2 + x_2,$$

$$(5c) \quad x_1x_2x_3,$$

typed as

```
\begin{subequations}\label{E:gp}
  \begin{gather}
    x_{\{1\}} x_{\{2\}} + x_{\{1\}}^{\{2\}} x_{\{2\}}^{\{2\}} + x_{\{3\}}, \\
    \label{E:gp1}\\
    x_{\{1\}} x_{\{3\}} + x_{\{1\}}^{\{2\}} x_{\{3\}}^{\{2\}} + x_{\{2\}}, \\
    \label{E:gp2}\\
    x_{\{1\}} x_{\{2\}} x_{\{3\}}, \label{E:gp3}
  \end{gather}
\end{subequations}
```

Then `\eqref{E:gp}` references the whole group of equations as (5), while

`\eqref{E:gp1}`, `\eqref{E:gp2}`, and `\eqref{E:gp3}`

reference the individual formulas as (5a), (5b), and (5c).

9.5 Aligned columns

The lines of multiline formulas are naturally divided into columns. In this section, we discuss how to typeset such formulas with *aligned columns*. All of these constructs are implemented with the `align` math environment and its variants.

In Section 1.7.3, you saw two simple, one-column examples of aligned columns—which we called *simple alignment*—and a special case of aligned columns—which we called *annotated alignment*.

The `align` environment can also create multiple aligned columns. The number of columns is restricted only by the width of the page. In the following example, there are two aligned columns:

$$(6) \quad \begin{array}{ll} f(x) = x + yz & g(x) = x + y + z \\ h(x) = xy + xz + yz & k(x) = (x + y)(x + z)(y + z) \end{array}$$

typed as

```
\begin{align}\label{E:mm3}
  f(x) &= x + yz & g(x) &= x + y + z \\
  h(x) &= xy + xz + yz & k(x) &= (x + y)(x + z)(y + z)
\end{align}
```

```
\notag
\end{align}
```

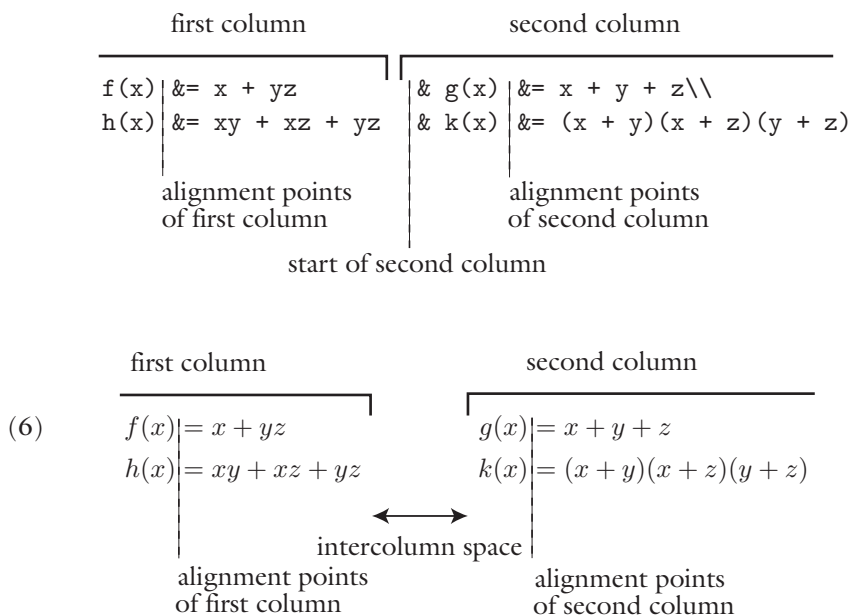


Figure 9.2: Two aligned columns: source and typeset.

Use Figure 9.2 to visualize how the alignment points in the source turn into alignment points in the typeset formula and the role played by the intercolumn space. Remember that the visual layout of the source is for your benefit only.

In a multicolumn `align` environment, the ampersand (`&`) plays two roles. It is a mark for the *alignment point* and it is also a *column separator*. In the line

```
f(x) &= x + yz          & g(x) &= x + y + z
```

the two columns are

1. $f(x) = x + yz$

2. $g(x) = x + y + z$

In each column, we use a single ampersand to mark the alignment point. Of the three `&` symbols in the previous example:

- The first $\&$ marks the *alignment point* of the first column.
- The second $\&$ is a *column separator* that separates the first and second columns.
- The third $\&$ marks the *alignment point* of the second column.

I use the convention of typing a space on the left of an alignment point $\&$ and no space on the right, and of putting spaces on both sides of $\&$ as a column separator.

If the number of columns is three, then there should be five $\&$'s in each line. Even-numbered $\&$'s are column separators and odd-numbered $\&$'s are alignment marks.



Rule: Ampersands

If there are n aligned columns, then each line should have at most $2n - 1$ ampersands. Even-numbered $\&$'s are column separators; odd-numbered $\&$'s mark the alignment points.

So for a single aligned column, you have to place one alignment point for each line. For two aligned columns, you have to place at most three alignment points for each line. The beginning of the line to the second $\&$ is the first column, then from the second $\&$ to the end of the line is the second column. Each line of each column has an alignment point marked by $\&$.

A column in a line may be empty—a gap is produced—or a line may have only a few columns. Both of these are illustrated by

$$\begin{array}{ccc}
 a_1 & & c_1 \\
 & b_2 & c_2 \\
 a_3 & &
 \end{array}$$

typed as

```

\begin{align*}
& a_1 & \& & \& & c_1 \\
& & \& b_2 & \& c_2 \\
& a_3
\end{align*}

```

9.5.1 An align variant

A variant of `align` is the flush alignment environment `flalign`, which moves the leftmost column as far left and the rightmost column as far right as space allows, making more room for the formula. Here is formula (6) again, followed by the `flalign`

variant:

$$(6) \quad \begin{array}{ll} f(x) = x + yz & g(x) = x + y + z \\ h(x) = xy + xz + yz & k(x) = (x + y)(x + z)(y + z) \end{array}$$

$$(7) \quad \begin{array}{ll} f(x) = x + yz & g(x) = x + y + z \\ h(x) = xy + xz + yz & k(x) = (x + y)(x + z)(y + z) \end{array}$$

The variant is typed as follows:

```
\begin{flalign}\label{E:mm3fl1}
  f(x) &= x + yz && & g(x) &= x + y + z \\
  h(x) &= xy + xz + yz && & k(x) &= (x + y)(x + z)(y + z) \\
  &\notag \\
\end{flalign}
```

9.5.2 eqnarray, *the ancestor of* align

L^AT_EX's original aligned math environment is eqnarray. Here is an example:

```
\begin{eqnarray}
  x &= & 17y \\
  y &> & a + b + c \\
\end{eqnarray}
```

which typesets as

$$(8) \quad \begin{array}{rcl} x & = & 17y \\ (9) \quad y & > & a + b + c \end{array}$$

You can type the same formulas with align:

```
\begin{align}
  x &= & 17y \\
  y &> & a + b + c \\
\end{align}
```

which typesets as

$$(10) \quad x = 17y$$

$$(11) \quad y > a + b + c$$

In the eqnarray environment the spacing is based on the spacing of the columns rather than on the spacing requirements of the symbols.

I mention `eqnarray` not for historical reasons but also for a very practical one. Unfortunately, a large number of journal submissions still use this construct, and have to be recoded in the editorial offices.



Tip Be kind to your editor and do not use `eqnarray`.

9.5.3 The subformula rule revisited

Suppose that you want to align the formula

$$x_1 + y_1 + \left(\sum_i \binom{5}{i} + a^2 \right)^2$$

with

$$\left(\sum_i \binom{5}{i} + \alpha^2 \right)^2$$

so that the $+ a^2$ in the first formula aligns with the $+ \alpha^2$ in the second formula. You might try typing

```
\begin{align*}
x_{\{1\}} + y_{\{1\}} + \left( \sum_i \binom{5}{\{i\}}
&+ a^{\{2\}} \right)^{\{2\}} \backslash
\left( \sum_i \binom{5}{\{i\}} + \alpha^{\{2\}} \right)^{\{2\}}
\end{align*}
```

But when you typeset this formula, you get the message

! Extra }, or forgotten \right.

This alignment structure violates the subformula rule because \LaTeX cannot typeset

```
x_{\{1\}} + y_{\{1\}} + \left( \sum_i \binom{5}{\{i\}}
```

because it is not a subformula.

As another simple example, try to align the $+$ in $\binom{a+b}{2}$ with the $+$ in $x + y$:

```
\begin{align}
\binom{a + b}{2} \backslash
x + y
\end{align}
```

When typesetting this formula, you get the message

! Missing } inserted.

Again, \LaTeX cannot typeset `\binom{a}{b}` because it is not a subformula.

To align the two formulas in the first example, add a `\phantom` command to push the second line to the right:

```
\begin{align*}
& x_{1} + y_{1} + \left( \sum_{i < 5} \binom{5}{i} \right. \\
& \quad \left. + a^{2} \right)^{2} \\
& \phantom{x_{1} + y_{1} + } \left( \sum_{i < 5} \binom{5}{i} + \alpha^{2} \right)^{2} \\
\end{align*}
```

yielding

$$x_1 + y_1 + \left(\sum_{i < 5} \binom{5}{i} + a^2 \right)^2$$

$$\left(\sum_{i < 5} \binom{5}{i} + \alpha^2 \right)^2$$

9.5.4 The alignat environment

Another variant of the `align` environment is the `alignat` environment, which is one of the most important alignment environments. While the `align` environment calculates how much space to put between the columns, the `alignat` environment leaves spacing up to the user. It is important to note that the `alignat` environment has a required argument, the number of columns.

Here is formula (6) typed with the `alignat` environment:

```
\begin{alignat}{2}\label{E:mm3A}
f(x) &= x + yz & g(x) &= x + y + z \\
h(x) &= xy + xz + yz & k(x) &= (x + y)(x + z)(y + z) \\
\end{alignat}
```

which typesets as

$$(12) \quad \begin{array}{ll} f(x) = x + yz & g(x) = x + y + z \\ h(x) = xy + xz + yz & k(x) = (x + y)(x + z)(y + z) \end{array}$$

This attempt did not work very well because `alignat` did not separate the two formulas in the second line. So you must provide the intercolumn spacing. For instance, if you want a `\quad` space between the columns, as in

$$(13) \quad \begin{array}{ll} f(x) = x + yz & g(x) = x + y + z \\ h(x) = xy + xz + yz & k(x) = (x + y)(x + z)(y + z) \end{array}$$

then type the formula as

```
\begin{alignat}{2}\label{E:mm3B}
  f(x) &= x + yz & g(x) &= x + y + z\\
  h(x) &= xy + xz + yz \quad & k(x) &= (x+y)(x+z)(y+z) \\
  \notag \\
\end{alignat}
```

The `alignat` environment is especially appropriate when annotating formulas where you would normally want a `\quad` between the formula and the text. To obtain

$$\begin{aligned}
 (14) \qquad x &= x \wedge (y \vee z) && \text{(by distributivity)} \\
 &= (x \wedge y) \vee (x \wedge z) && \text{(by condition (M))} \\
 &= y \vee z
 \end{aligned}$$

type

```
\begin{alignat}{2}\label{E:mm4}
  x &= x \wedge (y \vee z) & & \\
  &\quad \text{(by distributivity)} & & \\
  &= (x \wedge y) \vee (x \wedge z) & & \\
  &\quad \text{(by condition (M))} & & \\
  &= y \vee z & & \\
\end{alignat}
```

The `alignat` environment is very important for typing systems of equations such as

$$\begin{aligned}
 (15) \qquad (A + BC)x + Cy &= 0, \\
 (16) \qquad Ex + (F + G)y &= 23.
 \end{aligned}$$

typed as follows:

```
\begin{alignat}{2}
  (A + B C)x &+{} & C & & y = 0, \\
  Ex &+{} & (F + G)y &= 23. \\
\end{alignat}
```

Note again `+{}`. See also the subformula rule in Section 9.4.2.

As a last example, consider

$$\begin{aligned}
 (17) \qquad a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= y_1, \\
 (18) \qquad a_{21}x_1 + a_{22}x_2 &+ a_{24}x_4 = y_2, \\
 (19) \qquad a_{31}x_1 &+ a_{33}x_3 + a_{34}x_4 = y_3.
 \end{aligned}$$

typed as

```

\begin{alignat}{4}
a_{11}x_1 &+ a_{12}x_2 &&+ a_{13}x_3 &&
&&= y_1, \\
a_{21}x_1 &+ a_{22}x_2 && &&+ a_{24}x_4 \\
&&= y_2, \\
a_{31}x_1 &&&+ a_{33}x_3 &&+ a_{34}x_4 \\
&&= y_3.
\end{alignat}

```

Note that the argument of `alignat` does not have to be precise. If you want two columns, the argument can be 2, or 3, or any larger number. If you want to, you can simply type 10 and just ignore the argument. You may define a new environment (see Section 15.2.1) that does just that.

9.5.5 Inserting text

The `\intertext` command places one or more lines of text in the middle of an aligned environment. For instance, to obtain

┌

$$(20) \quad h(x) = \int \left(\frac{f(x) + g(x)}{1 + f^2(x)} + \frac{1 + f(x)g(x)}{\sqrt{1 - \sin x}} \right) dx$$

The reader may find the following form easier to read:

$$= \int \frac{1 + f(x)}{1 + g(x)} dx - 2 \arctan(x - 2)$$

└

you would type

```

\begin{align}\label{E:mm5}
h(x) &= \int \left( \frac{f(x) + g(x)}{1 + f^2(x)} + \right. \\
&\quad \left. \frac{1 + f(x)g(x)}{\sqrt{1 - \sin x}} \right) \\
&\quad \left. \right) dx \\
\intertext{The reader may find the following form}
&\quad \text{easier to read:} \\
&= \int \frac{1 + f(x)}{1 + g(x)} \\
&\quad dx - 2 \arctan(x - 2) \notag
\end{align}

```

Notice how the equal sign in the first formula is aligned with the equal sign in the second formula even though a line of text separates the two.

Here is another example, this one using `align*`:

┌

$$f(x) = x + yz \qquad g(x) = x + y + z$$

The reader may also find the following polynomials useful:

$$h(x) = xy + xz + yz \qquad k(x) = (x + y)(x + z)(y + z)$$

└

is typed as

```
\begin{align*}
  f(x) &= x + yz & \qquad g(x) &= x + y + z \\
  \intertext{The reader may also find the following}
  \text{polynomials useful:} \\
  h(x) &= xy + xz + yz \\
  &\qquad \qquad \qquad & \qquad \qquad k(x) &= (x + y)(x + z)(y + z)
\end{align*}
```

The `\intertext` command must follow a line separator command, `\\` or `*` (see Section 9.9). If you violate this rule, you get the message

```
! Misplaced \noalign. \intertext #1->\noalign
                        {\penalty \postdisplaypenalty
                        \vskip ...}
```

The text in `\intertext` can be centered using the `\centering` command (see Section 4.3).

9.6 Aligned subsidiary math environments

A *subsidiary math environment* is a math environment that can only be used *inside another math environment*. Think of it as creating a “large math symbol”.

In this section, we discuss aligned subsidiary math environments. We discuss adjusted subsidiary math environments, including cases, in Section 9.7.

9.6.1 Subsidiary variants

The `align`, `alignat`, and `gather` environments (see Sections 9.5, 9.5.4, and 9.2) have subsidiary versions. They are called `aligned`, `alignedat`, and `gathered`. To obtain

$$\begin{array}{rcl}
 x = 3 + \mathbf{p} + \alpha & & \mathbf{p} = 5 + a + \alpha \\
 y = 4 + \mathbf{q} & \text{using} & \mathbf{q} = 12 \\
 z = 5 + \mathbf{r} & & \mathbf{r} = 13 \\
 u = 6 + \mathbf{s} & & \mathbf{s} = 11 + d
 \end{array}$$

type

```

\[
\begin{aligned}
x &= 3 + \mathbf{p} + \alpha\\
y &= 4 + \mathbf{q}\\
z &= 5 + \mathbf{r}\\
u &= 6 + \mathbf{s}
\end{aligned}
\text{\quad using\quad}
\begin{gathered}
\mathbf{p} = 5 + a + \alpha\\
\mathbf{q} = 12\\
\mathbf{r} = 13\\
\mathbf{s} = 11 + d
\end{gathered}
\]
```

Note how the list of aligned formulas

$$\begin{array}{l}
 x = 3 + p + \alpha \\
 y = 4 + \mathbf{q} \\
 z = 5 + \mathbf{r} \\
 u = 6 + \mathbf{s}
 \end{array}$$

and the list of centered formulas

$$\begin{array}{c}
 \mathbf{p} = 5 + a + \alpha \\
 \mathbf{q} = 12 \\
 \mathbf{r} = 13 \\
 \mathbf{s} = 11 + d
 \end{array}$$

are treated as individual large symbols.

The `aligned`, `alignedat`, and `gathered` subsidiary math environments follow the same rules as `align` and `gather`. The `aligned` subsidiary math environment

allows any number of columns, but you must specify the intercolumn spacing as in the `alignat` environment.

You can use the `aligned` subsidiary math environment to rewrite formula (5) from Section 1.7.3 so that the formula number is centered between the two lines:

$$(21) \quad \begin{aligned} h(x) &= \int \left(\frac{f(x) + g(x)}{1 + f^2(x)} + \frac{1 + f(x)g(x)}{\sqrt{1 - \sin x}} \right) dx \\ &= \int \frac{1 + f(x)}{1 + g(x)} dx - 2 \arctan(x - 2) \end{aligned}$$

this is typed as

```
\begin{equation}\label{E:mm6}
\begin{aligned}
h(x) &= \int \left( \frac{f(x) + g(x)}{1 + f^2(x)} + \right. \\
&\quad \left. \frac{1 + f(x)g(x)}{\sqrt{1 - \sin x}} \right) dx \\
&= \int \frac{1 + f(x)}{1 + g(x)} dx - 2 \arctan (x - 2)
\end{aligned}
\end{equation}
```

See Section 9.6.2 for a better way to split a long formula.

Symbols, as a rule, are vertically centrally aligned. This is not normally an issue with math symbols, but it may be important with large symbols created by subsidiary math environments. The subsidiary math environments, `aligned`, `gathered`, and `array`, take `c`, `t`, or `b` as optional arguments to force vertically centered, top, or bottom alignment, respectively. The default is `c` (centered). To obtain

$$\begin{array}{ll} x = 3 + \mathbf{p} + \alpha & \mathbf{p} = 5 + a + \alpha \\ y = 4 + \mathbf{q} & \mathbf{q} = 12 \\ z = 5 + \mathbf{r} & \mathbf{r} = 13 \\ u = 6 + \mathbf{s} & \text{using } \mathbf{s} = 11 + d \end{array}$$

for example, you would type

```
\[
\begin{aligned}[b]
x &= 3 + \mathbf{p} + \alpha \\
y &= 4 + \mathbf{q} \\
z &= 5 + \mathbf{r} \\
&\text{using } \mathbf{s} = 11 + d
\end{aligned}
```



```

        u &= 6 + \mathbf{s}
\end{aligned}
\text{\quad using\quad}
\begin{gathered}[b]
    \mathbf{p} = 5 + a + \alpha\\
    \mathbf{q} = 12\\
    \mathbf{r} = 13\\
    \mathbf{s} = 11 + d
\end{gathered}
\]
```

There is no numbering or `\tag`-ing allowed in subsidiary math environments because \LaTeX does not number or tag what it considers to be a single symbol.

9.6.2 *Split*

The `split` subsidiary math environment is used to split a long formula into aligned parts. There are two major advantages to the use of the `split` environment:

1. The math environment that contains it considers the `split` environment to be a single equation, so it generates only one number for it.
2. If a `split` environment appears inside an `align` environment, the alignment point of the `split` environment is recognized by `align` and is used in aligning all the formulas in the `align` environment.

To illustrate the first, consider

$$(22) \quad (x_1x_2x_3x_4x_5x_6)^2 + (x_1x_2x_3x_4x_5 + x_1x_3x_4x_5x_6 + x_1x_2x_4x_5x_6 + x_1x_2x_3x_5x_6)^2$$

typed as

```

\begin{equation}\label{E:mm7}
\begin{split}
(x_{\{1\}x_{\{2\}}\&x_{\{3\}x_{\{4\}x_{\{5\}x_{\{6\}}\}^{\{2\}}\\
&+ (x_{\{1\}x_{\{2\}x_{\{3\}x_{\{4\}x_{\{5\}}\}
+ x_{\{1\}x_{\{3\}x_{\{4\}x_{\{5\}x_{\{6\}}\}
+ x_{\{1\}x_{\{2\}x_{\{4\}x_{\{5\}x_{\{6\}}\}
+ x_{\{1\}x_{\{2\}x_{\{3\}x_{\{5\}x_{\{6\}}\}^{\{2\}}
\end{split}
\end{equation}
```

See also the two examples of `split` in the `art2.tex` sample article in Section 11.3 and in the `samples` folder (see page 5).

To illustrate the second, here is an example of a `split` subsidiary math environment within an `align` environment:

$$(23) \quad \begin{aligned} f &= (x_1 x_2 x_3 x_4 x_5 x_6)^2 \\ &= (x_1 x_2 x_3 x_4 x_5 + x_1 x_3 x_4 x_5 x_6 + x_1 x_2 x_4 x_5 x_6 + x_1 x_2 x_3 x_5 x_6)^2, \end{aligned}$$

$$(24) \quad g = y_1 y_2 y_3.$$

which is typed as

```
\begin{align}\label{E:mm8}
  \begin{split}
    f &= (x_{1} x_{2} x_{3} x_{4} x_{5} x_{6})^2 \\
    &= (x_{1} x_{2} x_{3} x_{4} x_{5} \\
    &\quad + x_{1} x_{3} x_{4} x_{5} x_{6} \\
    &\quad + x_{1} x_{2} x_{4} x_{5} x_{6} \\
    &\quad + x_{1} x_{2} x_{3} x_{5} x_{6})^2, \\
  \end{split} \\
  g &= y_{1} y_{2} y_{3}.\label{E:mm9}
\end{align}
```

Notice the `\\` command following `\end{split}` to separate the lines for align.



Rule: split subsidiary math environment

1. `split` can only be used inside another math environment, such as `displaymath`, `equation`, `align`, `gather`, `flalign`, `gathered` and their *-ed variants.
2. A `split` formula has only one number, generated by L^AT_EX, or one tag from a `\tag` command. Use the `\notag` command to suppress numbering.
3. The `\label`, `\tag`, or `\notag` command must precede `\begin{split}` or follow `\end{split}`.

Here is an example of `split` inside a `gather` environment:

```
\begin{gather}\label{E:mm10}
  \begin{split}
    f &= (x_{1} x_{2} x_{3} x_{4} x_{5} x_{6})^2 \\
    &= (x_{1} x_{2} x_{3} x_{4} x_{5} \\
    &\quad + x_{1} x_{3} x_{4} x_{5} x_{6} \\
    &\quad + x_{1} x_{2} x_{4} x_{5} x_{6} \\
    &\quad + x_{1} x_{2} x_{3} x_{5} x_{6})^2 \\
    &= (x_{1} x_{2} x_{3} x_{4}
  \end{split}
\end{gather}
```

```

+ x_{1} x_{2} x_{3} x_{5}
+ x_{1} x_{2} x_{4} x_{5}
+ x_{1} x_{3} x_{4} x_{5})^{\{2\}}
\end{split}\\
\begin{aligned}
g &= y_{1} y_{2} y_{3}\\
h &= z_{1}^{\{2\}} z_{2}^{\{2\}} z_{3}^{\{2\}} z_{4}^{\{2\}}
\end{aligned}
\end{gather}

```

which produces

$$\begin{aligned}
 f &= (x_1 x_2 x_3 x_4 x_5 x_6)^2 \\
 (25) \quad &= (x_1 x_2 x_3 x_4 x_5 + x_1 x_3 x_4 x_5 x_6 + x_1 x_2 x_4 x_5 x_6 + x_1 x_2 x_3 x_5 x_6)^2 \\
 &= (x_1 x_2 x_3 x_4 + x_1 x_2 x_3 x_5 + x_1 x_2 x_4 x_5 + x_1 x_3 x_4 x_5)^2 \\
 g &= y_1 y_2 y_3 \\
 h &= z_1^2 z_2^2 z_3^2 z_4^2
 \end{aligned}$$



Tip Do not use `split` outside a displayed math environment.

If you try to use it, you get the message

`! Package amsmath Error: \begin{split} won't work here.`

You may want to read the discussion of the AMS document classes and `amsmath` package options in Section 11.5 that modify the placement of equation numbers.

9.7 Adjusted columns

In an *adjusted* multiline math environment, the columns are adjusted so that they are displayed centered, flush left, or flush right, instead of aligned (as in Section 9.5). Since you have no control line by line over the alignment of the columns, `&` has only one role to play—it is the column separator.

In Sections 9.2 and 9.3, we discussed two adjusted one-column math environments, `gather` and `multline`. All the other adjusted constructs are subsidiary math environments. For example, a `matrix` environment (see Section 9.7.1) produces a multicolumn centered display:

$$\begin{pmatrix} a+b+c & uv & x-y & 27 \\ a+b & u+v & z & 1340 \end{pmatrix} = \begin{pmatrix} 1 & 100 & 115 & 27 \\ 201 & 0 & 1 & 1340 \end{pmatrix}$$

The `array` environment (see Section 9.7.2) produces a multicolumn adjusted display:

$$\left(\begin{array}{cccc} a+b+c & uv & x-y & 27 \\ a+b & u+v & z & 1340 \end{array} \right) = \left(\begin{array}{cccc} 1 & 100 & 115 & 27 \\ 201 & 0 & 1 & 1340 \end{array} \right)$$

The columns are centered, flush left, or flush right. In this example, the first matrix has three centered columns and one flush right column, while the second matrix has four flush right columns. A variant, `cases` (see Sections 1.7.4 and 9.7.3), produces two columns set flush left:

$$(26) \quad f(x) = \begin{cases} -x^2, & \text{if } x < 0; \\ \alpha + x, & \text{if } 0 \leq x \leq 1; \\ x^2, & \text{otherwise.} \end{cases}$$

9.7.1 Matrices

Use the `matrix` subsidiary math environment to typeset matrices. For example,

```
\begin{equation*}
\left(
\begin{matrix}
a + b + c & uv & x - y & 27 \\
a + b & u + v & z & 1340
\end{matrix}
\right) =
\left(
\begin{matrix}
1 & 100 & 115 & 27 \\
201 & 0 & 1 & 1340
\end{matrix}
\right)
\end{equation*}
```

produces

$$\left(\begin{array}{cccc} a+b+c & uv & x-y & 27 \\ a+b & u+v & z & 1340 \end{array} \right) = \left(\begin{array}{cccc} 1 & 100 & 115 & 27 \\ 201 & 0 & 1 & 1340 \end{array} \right)$$

If you use `matrix` on its own, that is, outside a math environment,

```
\begin{matrix}
a + b + c & uv & x - y & 27 \\
a + b & u + v & z & 134
\end{matrix}
```

you get the message

```
! Missing $ inserted.
<inserted text>
$
1.5 \begin{matrix}
```

obliquely reminding you that `matrix` is a subsidiary math environment.

The `matrix` subsidiary math environment provides a matrix of up to 10 centered columns. If you need more columns, you have to ask for them. The following example sets the number of columns to 12:

```
\begin{equation}\label{E:mm12}
\setcounter{MaxMatrixCols}{12}
\begin{matrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
1 & 2 & 3 & \hdotsfor{7} & & & & & & & 11 & 12
\end{matrix}
\end{equation}
```

produces

$$(27) \quad \begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 1 & 2 & 3 & \dots\dots\dots & & & & & & & 11 & 12 \end{array}$$

We discuss `\setcounter` and other counters further in Section 15.5.1.

You can have dots span any number of columns with the `\hdotsfor` command, as in (27). The argument of this command specifies the number of columns to fill (which is one more than the number of `&`'s the command replaces). The `\hdotsfor` command must either appear at the beginning of a row or immediately following an ampersand (`&`). If you violate this rule, you get the message

```
! Misplaced \omit.
\multispan #1->\omit
\mscount #1\relax \loop \ifnum
\mscount ...
1.12 \end{equation}
```

The `\hdotsfor` command also takes an optional argument, a number that multiplies the spacing between the dots. The default is 1. For instance, if we replace `\hdotsfor{7}` in the previous example by `\hdotsfor[3]{7}`, then we get

$$(28) \quad \begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 1 & 2 & 3 & \dots\dots\dots & & & & & & & 11 & 12 \end{array}$$

We can replace a part of a matrix column with a large symbol.

$$a = \begin{pmatrix} (a_{11}) \\ \cdots \\ (a_{n1}) \end{pmatrix} \mathbf{0}, \quad (a_{k1}) = \begin{pmatrix} 0 \dots 0 & 1 & 0 \dots 0 \\ & 0 & \\ \mathbf{0} & \cdots & \mathbf{0} \\ & 0 & \end{pmatrix}$$

typed as

```
\newcommand{\BigFig}[1]{\parbox{12pt}{\Huge #1}}
\newcommand{\BigZero}{\BigFig{0}}
\[
a=\left( \begin{matrix}
(a_{11})\\
\cdots & \BigZero \\
(a_{n1})
\end{matrix} \right) \mathbf{0}, \quad
(a_{k1})=\left( \begin{matrix}
0\dots 0 & 1 & 0\dots 0 \\
& 0 & \\
\mathbf{0} & \cdots & \mathbf{0} \\
& 0 &
\end{matrix} \right)
\]
```

Matrix variants

A matrix may be enclosed by delimiters (see Section 7.5.1) in a number of different ways:

$$\begin{array}{ccc} \begin{matrix} a+b+c & uv \\ a+b & c+d \end{matrix} & \begin{pmatrix} a+b+c & uv \\ a+b & c+d \end{pmatrix} & \begin{bmatrix} a+b+c & uv \\ a+b & c+d \end{bmatrix} \\ \left| \begin{matrix} a+b+c & uv \\ a+b & c+d \end{matrix} \right| & \left\| \begin{matrix} a+b+c & uv \\ a+b & c+d \end{matrix} \right\| & \left\{ \begin{matrix} a+b+c & uv \\ a+b & c+d \end{matrix} \right\} \end{array}$$

The first matrix is typed as

```
\begin{matrix}
a + b + c & & uv \\
a + b & & c + d
\end{matrix}
```

The others are typed in the same way, except that they use the `pmatrix`, `bmatrix`, `vmatrix`, `Vmatrix`, and `Bmatrix` environments, respectively. We can use other delimiters, as in

```
\begin{equation*}
\left(
\begin{matrix}
1 & & 0 & & \dots & & 0 \\
0 & & 1 & & \dots & & 0 \\
\vdots & & \vdots & & \ddots & & \vdots \\
0 & & 0 & & \dots & & 1
\end{matrix}
\right)
\end{equation*}
```

which produces

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

This example also uses *vertical dots* provided by the `\vdots` commands and *diagonal dots* provided by the `\ddots` commands.

Small matrix

If you put a matrix in an inline math formula, it may be too large. Instead, use the `smallmatrix` environment. Compare $\begin{pmatrix} a+b+c & uv \\ a+b & c+d \end{pmatrix}$, typed as

```
\begin{pmatrix}
a + b + c & uv \\
a + b & c + d
\end{pmatrix}
```

with the small matrix $\begin{pmatrix} a+b+c & uv \\ a+b & c+d \end{pmatrix}$, typed as

```
\left(
\begin{smallmatrix}
a + b + c & uv \\
a + b & c + d
\end{smallmatrix}
\right)
```

There are no delimited variants of `smallmatrix` similar to those of `matrix`. Instead, use the `\left` and `\right` commands with delimiters to enclose a small matrix. The `\hdotsfor` command does not work in a small matrix.

9.7.2 Arrays

The array subsidiary math environment is a variant of `matrix`. For array, you must specify the alignment of each column, and you have more options to customize it.

The first matrix in the introduction to Section 9.7 would be typed as follows using the array subsidiary math environment:

```
\begin{equation*}
\left(
\begin{array}{cccc}
a + b + c & uv & x - y & 27 \\
a + b & u + v & z & 134
\end{array}
\right)
\end{equation*}
```

which produces

$$\left(\begin{array}{cccc} a + b + c & uv & x - y & 27 \\ a + b & u + v & z & 134 \end{array} \right)$$



Rule: array subsidiary math environment

1. Adjacent columns are separated by an ampersand (&).
2. The argument of `\begin{array}` is mandatory. The argument is a series of the letters l, r, or c, signifying that the corresponding column in the array should be set flush left, flush right, or centered, respectively.

The matrix

$$\left(\begin{array}{cccc} a + b + c & uv & x - y & 27 \\ a + b & u + v & z & 134 \end{array} \right)$$

could not have been typeset with `matrix` since the last column is set flush right. This is not quite true, of course. In a `matrix` environment, `\hfill 27` would force the number 27 to be set flush right (see Section 3.7.4).

If the argument of `\begin{array}` is missing, as in

```
\begin{equation}
\begin{array}
a + b + c & uv & x - y & 27 \\
a + b & u + v & z & 134
\end{array}
\end{equation}
```

L^AT_EX generates the message


```

2 & 1 & -1 & -1 \\
2 & 2 & 1 & 0
\end{array}
\]
```

In Section 9.7.1 we have the matrix example:

$$a = \begin{pmatrix} (a_{11}) \\ \cdots \\ (a_{n1}) \end{pmatrix} \mathbf{0}$$

If rows are spanned, we need to use array instead of matrix:

$$\begin{bmatrix} a & b & \mathbf{0} \\ c & d & \\ \mathbf{0} & m & n \\ & k & l \end{bmatrix}$$

typed as (the `\BigZero` command is defined on page 239)

```

\left[ \hspace{-\arraycolsep}
  spacing is automatic with matrix but not with array
\begin{array}{ccc}
a & b & \multicolumn{2}{c}{\hspace{1.5ex}\BigZero}\\
c & d & \multicolumn{2}{c}{\hspace{1.5ex}\BigZero}& m & n \\
\multicolumn{2}{c}{\hspace{1.5ex}\BigZero}& k & l
\end{array}
\hspace{-\arraycolsep} \right]
\end{equation*}
```

9.7.3 Cases

The cases environment is also a subsidiary math environment. Here is the example from Section 1.7.4 and the introduction to this section:

$$f(x) = \begin{cases} -x^2, & \text{if } x < 0; \\ \alpha + x, & \text{if } 0 \leq x \leq 1; \\ x^2, & \text{otherwise.} \end{cases}$$

It is typed as

```

\begin{equation}
  f(x)=
  \begin{cases}
    -x^2, & \&\text{if } x < 0;\}\\
    \alpha + x, & \&\text{if } 0 \leq x \leq 1;\}\\
    x^2, & \&\text{otherwise.}
  \end{cases}
\end{equation}

```

It would be easy to code the cases environment as a special case of the array subsidiary math environment:

```

\begin{equation}
  f(x) =
  \left\{
    \begin{array}{ll}
      -x^2, & \&\text{if } x < 0;\}\\
      \alpha + x, & \&\text{if } 0 \leq x \leq 1;\}\\
      x^2, & \&\text{otherwise.}
    \end{array}
  \right.
\end{equation}

```

or of the alignedat subsidiary math environment:

```

\begin{equation*}
  f(x) =
  \left\{
    \begin{alignedat}{2}
      &-x^2, &&\quad \&\text{if } x < 0;\}\\
      &\alpha + x, &&\quad \&\text{if } 0 \leq x \leq 1;\}\\
      &x^2, &&\quad \&\text{otherwise.}
    \end{alignedat}
  \right.
\end{equation*}

```

9.8 Commutative diagrams

The `amscd` package provides the CD subsidiary math environment for typesetting simple commutative diagrams. To use it, make sure that the command

```
\usepackage{amscd}
```

is in the preamble of the document.

For instance, to obtain

$$\begin{array}{ccc} A & \longrightarrow & B \\ \downarrow & & \downarrow \\ C & \xlongequal{\quad} & D \end{array}$$

type

```
\[
\begin{CD}
A @>>> B \\
@VVV @VVV \\
C @= D
\end{CD}
\]
```

A commutative diagram is a matrix made up of two kinds of rows, *horizontal rows*, that is, rows with horizontal arrows; and *vertical rows*, rows with vertical arrows. For example,

A @>>> B

is a typical horizontal row. It defines two columns and a connecting horizontal arrow @>>>. There may also be more than two columns, as in

A @>>> B @>>> C @= D @<<< E @<<< F

The connecting pieces can be:

- stretchable right arrows, @>>>,
- stretchable left arrows, @<<<,
- stretchable equal signs, @=,
- blanks, @. .

The label above a stretchable arrow should be typed between the first and second > or < symbols, whereas the label below should be typed between the second and third > or < symbols. You can have both.

The following is a typical vertical row containing vertical arrows:

@VVV @VVV @AAA

The vertical pieces could be:

- stretchable down arrows, @VVV,
- stretchable up arrows, @AAA,

- double vertical lines, @| or @\vert,
- blanks, @. .

The vertical arrows are placed starting with the first column.

The label to the left of a stretchable vertical arrow should be typed between the first and second V or A, whereas the label on the right should be typed between the second and third V or A symbols. You can have both.

These constructs are illustrated in

$$\begin{array}{ccccc}
 \mathbb{C} & \xrightarrow{H_1} & \mathbb{C} & \xrightarrow{H_2} & \mathbb{C} \\
 P_{c,3} \downarrow & & P_{\bar{c},3} \downarrow & & \downarrow P_{-c,3} \\
 \mathbb{C} & \xrightarrow{H_1} & \mathbb{C} & \xrightarrow{H_2} & \mathbb{C}
 \end{array}$$

typed as

```

\[
\begin{CD}
\mathbb{C} @>H_{1}>> \mathbb{C} @>H_{2}>> \mathbb{C} \\
@VP_{c,3}VV @VP_{\bar{c},3}VV @VVP_{-c,3}V \\
\mathbb{C} @>H_{1}>> \mathbb{C} @>H_{2}>> \mathbb{C}
\end{CD}
\]
```

Here is another example utilizing the \text command, followed by its source:

$$\begin{array}{ccccccc}
 A & \xrightarrow{\log} & B & \xrightarrow{\text{bottom}} & C & \xlongequal{\quad} & D \longleftarrow E \longleftarrow F \\
 \text{one-one} \downarrow & & & & \uparrow \text{onto} & & \parallel \\
 X & \xlongequal{\quad} & Y & \longrightarrow & Z & \longrightarrow & U \\
 \beta \uparrow & & \uparrow \gamma & & \downarrow & & \downarrow \\
 D & \xrightarrow{\alpha} & E & \longrightarrow & H & & I
 \end{array}$$

```

\[
\begin{CD}
A @>\log>> B @>>\text{bottom}>> C \\
@= @. @<<< @. E \\
@<<< @. F \\
@V\text{one-one}VV @. @AA\text{onto}A @| \\
X @= Y @>>> Z \\
@>>> @. U \\
@A\beta AA @AA\gamma A @VVV @VVV
\end{CD}
\]
```

```

D      @>\alpha>> E      @>>> H
      @.      I\\
\end{CD}
\]
```

Diagrams requiring more advanced commands should be done with a drawing (or drafting) application or with specialized packages. The AMS recommends Kristofer H. Rose and Ross Moore's `xy-pic` package. If you get familiar with the `TikZ` package in Chapter 14, then you should utilize the `tikzcd` package of Florêncio Neves to draw commutative diagrams; see Section 14.1.2.

9.9 *Adjusting the display*

By default, the math environments described in this chapter do not allow page breaks. While a page break in a `cases` environment is obviously not desirable, it may be acceptable in an `align` or `gather` environment. You can allow page breaks by using the

`\allowdisplaybreaks`

command. It allows page breaks in a multiline math environment within its scope. For instance,

```

{\allowdisplaybreaks
\begin{align}\label{E:mm13}
  a &= b + c, \\
  d &= e + f, \\
  x &= y + z, \\
  u &= v + w.
\end{align}
}% end of \allowdisplaybreaks
```

allows a page break after any one of the first three lines.

Within the scope of an `\allowdisplaybreaks` command, use the `*` command to prohibit a break after that line. The line separators `\\` and `*` can use an optional argument to add some additional interline space (see Section 3.6.2).

Just before the line separator command (`\\`), include a `\displaybreak` command to force a break, or a

`\displaybreak[0]`

command to allow one. `\displaybreak[n]`, where n is 1, 2, or 3, specifies the intermediate steps between allowing and forcing a break. `\displaybreak[4]` is the same as `\displaybreak`.

You can easily visualize these rules:

```
allow display break =
\displaybreak[0] \displaybreak[1] ... \displaybreak[4]
                                = \displaybreak
                                = force display break
```

Note the similarity between the `displaybreak` sequence and the `pagebreak` sequence in Section 3.6.3.

If you want to allow page breaks in all multiline math environments in your document, place the `\allowdisplaybreaks[1]` command in the preamble of your document. The optional argument can vary from 1 to 4, in order of increasing permissiveness.

Note that none of the subsidiary math environments are affected by any variant of the `\displaybreak` or the `\allowdisplaybreaks` commands.

PART V

Document Structure



Documents

In this chapter, we take up the organization of shorter documents. Longer documents and books are discussed in Part VIII.

If you are writing a *simple article*, start with a template (see Section 11.4), then you can safely ignore much of the material discussed in this chapter. In more complicated articles, you may need the material discussed in this chapter.

Section 10.1 discusses document structure in general; the preamble is presented in Section 10.2, and Section 10.3 discusses the top matter, in particular, the `abstract` environment. Section 10.4 presents the main matter, including sectioning, cross-referencing, tables, and figures. Section 10.5 covers the back matter, including the bibliography and index.

In Section 10.1–10.5, we discuss the logical design of a \LaTeX document. The visual design is largely left to the document class. In Section 10.6, however, we briefly discuss one frequently adjusted aspect of visual design, the page style.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-55281-6_10

10.1 The structure of a document

The source file of a \LaTeX document is divided into two main parts: the preamble and the body (see Figure 10.1).

Preamble This is the portion of the source file before the

```
\begin{document}
```

command. It contains definitions and instructions that affect the entire document.

Body This is the content of the document environment. It contains all the material to be typeset.

These statements oversimplify the situation somewhat. For instance, you can define a command in the preamble to typeset some text that will appear wherever the command is used in the body, but the text is actually typed in the preamble. Nevertheless, I hope the division between the preamble and the body is clear.

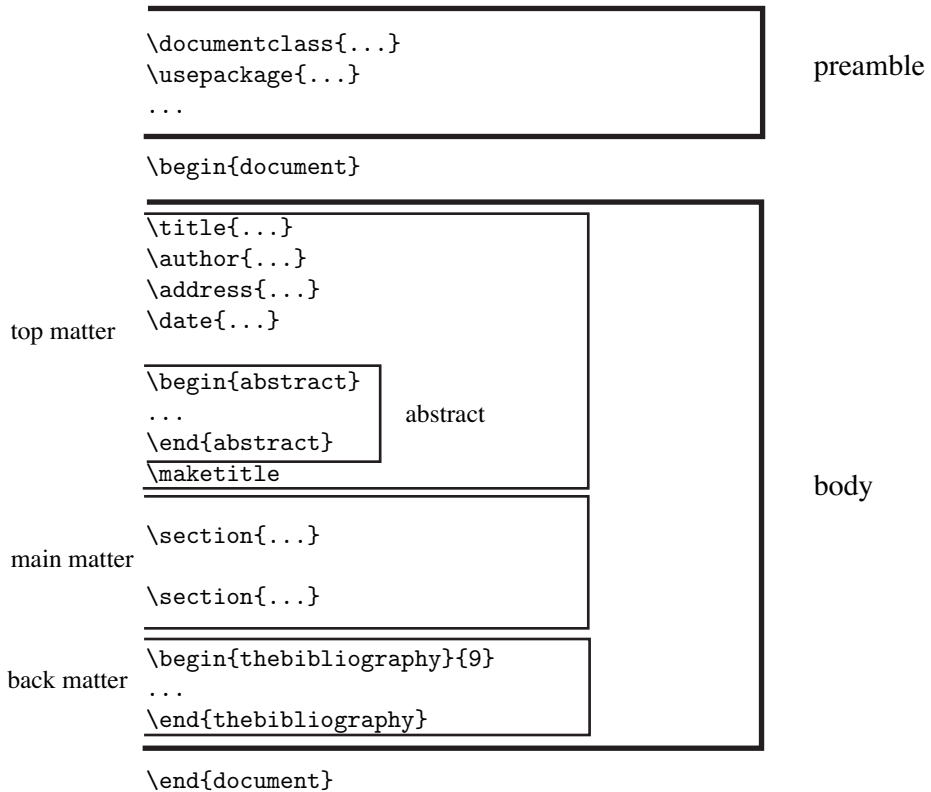


Figure 10.1: The structure of a \LaTeX document.

The body is divided into three parts:

Top matter This is the first part of the body. It is concluded with the `\maketitle` command. Traditionally it included only the `\title`, the `\author`, and the `\date` commands. The top matter is derived from these commands and from it the title page of an article was designed. This evolved to include a lot more information about the author(s), for instance, their e-mail addresses, academic affiliations, home pages; and about the article, for instance, research support, subject classification. The typeset top matter now is split into several locations, the top and bottom of the first page and the bottom of the last page. See pages 299–302 for an example and Section 19.1.2 for more components that can be used in longer documents and books.

Main matter This is the main part of the document, including any appendices.

Back matter This is the material that is typeset at the end of the document. For a typical shorter document, the back matter is just the bibliography. See Section 19.1.2 for more information about additional components—such as the index—that are often used in longer documents and books.

10.2 The preamble

You were introduced to the preamble of a document in Section 1.8. Recall that the preamble contains the crucial `\documentclass` line, specifying the document class and the options that modify its behavior. For instance,

```
\documentclass[draft,reqno]{amsart}
```

loads the document class `amsart` with the `draft` option, which paints a slug in the margin indicating lines that are too wide (see Section 3.6.1), and the `reqno` option, which places the equation numbers on the right (see Section 11.5).

`article` is the most popular legacy document class. The command

```
\documentclass[titlepage,twoside]{article}
```

loads the document class `article` with the `titlepage` option, which creates a separate title page and places the abstract on a separate page, and the `twoside` option, which formats the typeset article for printing on both sides of the paper.

The `\documentclass` command is usually followed by the `\usepackage` commands, which load \LaTeX enhancements called *packages*. For instance,

```
\usepackage{latexsym}
```

loads a package that defines some additional \LaTeX symbol names and

```
\usepackage[demo]{graphicx}
```

loads the `graphicx` package (see Section 10.4.3) with the `demo` option that inserts rectangles in place of the illustrations. Document class options are also passed on to the packages as possible options, so

```
\documentclass[demo]{amsart}
\usepackage{graphicx}
```

would also load the `graphicx` package with the `demo` option unless it is invoked with

```
\usepackage[final]{graphicx}
```

Any document class options that are not relevant for a package are ignored.

`\usepackage` commands can also be combined:

```
\usepackage{amssymb,latexsym}
```

is the same as

```
\usepackage{amssymb}
\usepackage{latexsym}
```

Document class files have a `cls` extension, whereas package files are designated by the `sty` extension. The document class `amsart` is defined in the `amsart.cls` file, whereas the `graphicx` package is defined in the `graphicx.sty` file. You may define your own packages, such as the `newlattice` package described in Section 15.3.

The preamble normally contains any custom commands (see Chapter 15) and the proclamation definitions (see Section 4.4). Some commands can only be in the preamble. `\DeclareMathOperator` is such a command (see Section 15.1.6) and so is `\numberwithin` (see Section 7.3). If you put such a command in the body, for example, `\DeclareMathOperator`, you get a message:

```
! LaTeX Error: Can be used only in preamble.
1.103 \DeclareMathOperator
```

There is a command that may only be placed *before* the

```
\documentclass{...}
```

line:

```
\NeedsTeXFormat{LaTeX2e}[2005/12/01]
```

This command checks the version of \LaTeX being used to typeset the document and issues a warning if it is older than December 1, 2005, or whatever date you specified. Use this optional date argument if your document contains a feature that was introduced on or after the date specified or if an earlier version had a bug that would materially affect the typesetting of your document.

For instance, if you use the `\textsubscript` command, introduced in the December 1, 2005 release, then you may use the `\NeedsTeXFormat` line shown above. \LaTeX now hardly changes from year to year, so this command is rarely used except in document class files or package files.

10.3 Top matter

The top matter of an article is part of the article body and, as a rule, it contains the material used to create the “title page” and, optionally, an abstract.

Discussion of the top matter should take place in the context of a particular document class. We discuss the top matter of the `amsart` document class in Section 1.8, and we continue discussing it in much more detail in Section 11.2.

Long documents, such as books, have rather complicated top matter such as tables of contents (see Chapter 19). In this section, we only discuss the abstract.

Abstract

Most standard document classes, except those for letters and books, make provision for an abstract, typed in an `abstract` environment.

The document class formats the heading as `ABSTRACT`, or some variant, and, as a rule, typesets the text of the abstract in smaller type with wider margins.

The `amsart` document class requires that you place the abstract environment *before* the `\maketitle` command (see Figure 10.1). See the abstract in the sample article `art1.tex` on page 4. If you forget to place it there, you get the warning

```
Class amsart Warning:
Abstract should precede \maketitle in AMS document classes;
reported on input line 20.
```

and the abstract is typeset wherever the abstract environment happens to be placed.

In the `article` document class you place the abstract *after* the `\maketitle` command. If you place the abstract before the `\maketitle` command, the abstract is placed on page 1, and the article starts on page 2.

If the abstract and the “footnotes” from the top matter fill the first page, the second page has no running head. To fix this, follow the `\maketitle` command with the `\clearpage` command (see Section 3.6.3).

10.4 Main matter

The main matter contains most of the essential parts of the document, including the appendices.

We discuss now how to structure the main matter. We describe sectioning in Section 10.4.1, cross-referencing in Section 10.4.2, and tables and figures in Section 10.4.3.

10.4.1 Sectioning

The main matter of a typical shorter document is divided into *sections*. We discuss sectioning of longer documents in Section 19.1.1.

Sections

L^AT_EX is instructed to start a section with the `\section` command, which takes the title of the section as its argument. This argument may also be used for the running head, and it is also placed in the table of contents (see Section 19.2), which means that you need to protect fragile commands with the `\protect` command (see Section 3.3.3). L^AT_EX automatically assigns a section number and typesets the section number followed by the section title.

Any `\section` command may be followed by a `\label` command, so that you can refer to the section number generated by L^AT_EX, as in

```
\section{Introduction}\label{S:intro}
```

The command `\ref{S:intro}` refers to the number of the section and the command `\pageref{S:intro}` refers to the number of the typeset page where the section title appears.



Tip You save a lot of work if in the source file you type in the cross-reference:

```
\section{Introduction}\label{S:intro}
%Section~\ref{S:intro}
```

Other sectioning commands

A section may be subdivided into *subsections*, which may themselves be divided into *subsubsections*, *paragraphs*, and *subparagraphs*. Subsections are numbered within a section (in Section 1, they are numbered 1.1, 1.2, and so on). Here is the whole hierarchy:

```
\section
  \subsection
    \subsubsection
      \paragraph
        \subparagraph
```

It is important to understand that the five levels of sectioning are not just five different styles for typesetting section headers, but they form a hierarchy. You should never have a subsection outside a section, a subsubsection outside a subsection, and so on. For instance, if the first sectioning command in your document is `\subsection`, the subsections are numbered 0.1, 0.2, Or if in the first section of your document the first sectioning command is `\subsubsection`, the subsubsections are numbered 1.0.1, 1.0.2, Both are clearly undesirable.

There are two additional sectioning commands provided by the `report` and by the book document classes (`book` and `amsbook`): `\chapter` and `\part` (discussed in Section 19.1.1).

Any sectioning command may be followed by a `\label` command so that you can refer to the number (if any) generated by L^AT_EX and the page on which it appears (see Section 10.4.2).

There is also the seldom used top level `\specialsection` command. Articles do not have parts and chapters, but sometimes a long article may require further division using the `\specialsection` command.

The form of sectioning commands

All sectioning commands take one of the following three forms, illustrated below with the `\section` command:

Form 1 The simplest form is

```
\section{title}
```

where *title* is the section title, of course. You need to protect any fragile commands in *title* with the `\protect` command (see Section 3.3.3).

Form 2 The sectioning command may have an optional argument

```
\section[short_title]{title}
```

The optional *short_title* argument is used in the running head. See Section 19.2 on what goes into the table of contents. Protect any fragile commands in *short_title* with the `\protect` command (see Section 3.3.3).

Form 3 Finally, we consider the *-ed version

```
\section*{title}
```

There are no section numbers printed and the *title* is not included in the running head. Remember that if you * a section, all subsections, and so on, must also be *-ed to avoid having strange section numbers.

Sectioning commands typeset

Consider the following text:

```
\section{Introduction}\label{S:Intro}
We shall discuss the main contributors of this era.
\subsection{Birkhoff's contributions}\label{S:contrib}
\subsubsection{The years 1935--1945}\label{S:1935}
Going to Oxford was a major step.
```

```
\paragraph{The first paper}
What should be the definition of a universal algebra?
\subparagraph{The idea}
One should read Whitehead very carefully.
```

This is how this looks typeset in the `amsart` document class:

┌

1 Introduction

We shall discuss the main contributors of this era.

1.1 Birkhoff's contributions

1.1.1 The years 1935–1945

Going to Oxford was a major step.

The first paper What should be the definition of a universal algebra?

The idea One should read Whitehead very carefully.

└

Notice that paragraphs and subparagraphs are not displayed prominently by the AMS .
By contrast, look at the same text typeset in the legacy `article` document class:

┌

1. INTRODUCTION

We shall discuss the main contributors of this era.

1.1. Birkhoff's contributions.

1.1.1. *The years 1935–1945.* Going to Oxford was a major step.

The first paper. What should be the definition of a universal algebra?

The idea. One should read Whitehead very carefully.

└

This illustrates vividly one huge difference between the two document classes, the visual handling of sectioning.

Section 15.5.1 discusses how you can change the format of the section numbers, and how to specify which sectioning levels are to be numbered.

Section 2.2 of LC3 explains how to change the layout of section headings, especially useful for document class designers.

Appendix

In the main matter, if the article contains appendices, mark the beginning of the appendices with the `\appendix` command. After the `\appendix` command, the `\section` command starts the appendices (for books; see Section 19.1.2):

```
\appendix
\section{A proof of the Main Theorem}\label{S:geom}
```

This produces Appendix A with the given title, typeset just like a section.

Note that appendices may be labeled and cross-referenced like any other section. In an appendix, subsections are numbered A.1, A.2, and so on, subsubsections within A.1 are numbered A.1.1, A.1.2, and so on.

Let me repeat, `\appendix` is not like `\section`. It is not a command with an argument. Appendices are named by arguments of the `\section`—commands (for books, by the `\chapter`—commands) *placed after* the `\appendix` command.

10.4.2 Cross-referencing

There are three types of cross-referencing available in \LaTeX :

1. Symbolic referencing with `\ref` and for equations `\eqref`.
2. Page referencing with `\pageref`.
3. Bibliographic referencing with `\cite`.

In this section, we discuss the first two, while bibliographies are discussed in Section 10.5.1 and in Chapter 16.

Symbolic referencing

Wherever \LaTeX can automatically generate a number in your document, you can place a `\label` command

```
\label{symbol}
```

Then, at any place in your document, you can use the `\ref` command

```
\ref{symbol}
```

to place that number in the document. We call *symbol* the *label*. You can use labels for sectioning units, equations, figures, tables, items in an enumerated list environment (see Section 4.2.1), as well as for theorems and other proclamations.

If the equation labeled `E:int` is the fifth equation in an article, then \LaTeX stores the number 5 for the label `E:int`, so `\ref{E:int}` produces the number 5. If equations are numbered within sections (see Section 7.3), and an equation is the third equation in Section 2, then \LaTeX stores the number 2.3 for the label `E:int`, so the reference `\ref{E:int}` produces the number 2.3.

Example 1 The present section starts with the command

```
\section{Main matter}\label{S:MainMatter}
```

So `\ref{S:MainMatter}` produces the number 10.4, and we get the number of the typeset page where the section title appears with `\pageref{S:MainMatter}`, which is 255.



Tip Type

```
\section{Main matter}\label{S:MainMatter}
%Section~\ref{S:MainMatter}
```

to make cross-referencing quicker.

Example 2

```
\begin{equation}\label{E:int}
\int_0^{\pi} \sin x \, dx = 2.
\end{equation}
```

In this case, `\ref{E:int}` produces the number of the equation, `\eqref{E:int}` produces the number of the equation in parentheses.



Tip Type

```
\begin{equation}\label{E:int}%\eqref{E:int}
\int_0^{\pi} \sin x \, dx = 2.
\end{equation}
```

to make cross-referencing quicker.



Tip If you have to reference an equation in the statement of a theorem, always use `\eqref`. Do not use `\eqref` to reference anything but equations. (See the `\itemref` command introduced in Section 15.1.2.)

Example 3

```
\begin{theorem}\label{T:fund}
  Statement of theorem.
\end{theorem}
```

The reference `\ref{T:fund}` produces the number of the theorem.

**Tip** Type

```
\begin{theorem}\label{T:fund}%Theorem~\ref{T:fund}
  Statement of theorem.
\end{theorem}
```

to make cross-referencing quicker.

**Tip** Remember to typeset a document twice to see a change in a cross-reference.

If you typeset only once, and \LaTeX suspects that the cross-references have not been updated, you get a warning:

```
LaTeX Warning: Label(s) may have changed.
Rerun to get cross-references right.
```

**Rule:** `\label` command

The argument of the `\label` command is a string of letters, punctuation marks, and digits. It is case sensitive, so `S:intro` is different from `S:Intro`.

**Rule:** `\label` command

Place a `\label` command immediately after the command that generates the number.

The following is not compulsory but advisable.



Tip When referencing:

see Section~\ref{S:Intro} proved in Theorem~\ref{T:main}
 or
 see Sections~\ref{S:Intro} and~\ref{S:main}
 use ties (~).

It cannot be overemphasized how useful automatic cross-referencing can be when writing a document.



Tip Make your labels meaningful *to yourself*, so *you* easily remember them.



Tip Systematize your labels. For example, start the label for a section with S:, theorem with T:, lemma with L:, and so on. Then you can have a main section with label S:main, a main theorem with label T:main, and so on.

When you are cross-referencing, even if you follow these tips, it may not be easy to remember a label. David Carlisle’s `showkeys` package may help you out. It is part of the tools distribution. Include the line

```
\usepackage{showkeys}
```

in the preamble of your document. The `showkeys` package shows all symbolic references in the margin of the typeset document. With the `notcite` option, my preference,

```
\usepackage[notcite]{showkeys}
```

`showkeys` does not show the labels for bibliographic references. When the document is ready for final typesetting, then comment out this line.

You may want to experiment with the `varioref` package, which provides references indicating the relative location of the referred item by adding “on the facing page”, “on the preceding page” or “on the following page”. Invoke it with

```
\usepackage[nospace]{varioref}
```

and use the command `\vref` instead of `\ref`. For more detail see Section 2.4.1 of LC3.

Absolute referencing

There are two forms of absolute referencing.

1. Equations can be *tagged*. The `\tag{name}` command attaches a name to the formula. The tag replaces the equation number.
2. Items in an `itemize` environment can be tagged with the `\item[name]` construct. The tag replaces the item number.

Our first example is the equation

(Int)
$$\int_0^{\pi} \sin x \, dx = 2$$

it is typed as

```
\begin{equation}
  \int_0^{\pi} \sin x \, dx = 2 \tag{Int}
\end{equation}
```

Our second example is the numbered list:

┌

This space has the following properties:

- (a) Grade 2 Cantor;
- (b) Half-smooth Hausdorff;
- (c) Metrizable smooth.

└

typed as

```
\noindent This space has the following properties:
\begin{enumerate}
  \item[(a)] Grade 2 Cantor\label{Cantor};
  \item[(b)] Half-smooth Hausdorff\label{Hausdorff};
  \item[(c)] Metrizable smooth\label{smooth}.
\end{enumerate}
```

Tags are *absolute*. This equation is *always* referred to as (Int). Equation numbers, on the other hand, are *relative*, that is, they may change when the file is edited.



Tip Do not label absolute references. It may lead to problems hard to explain.

Page referencing

The command

```
\pageref{symbol}
```

produces the number of the typeset page corresponding to the location of the command `\label{symbol}`. For example, if the following text is typeset on page 5,

```
There may be three types of problems with the
construction of such lattices.\label{problem}
```

and you type

```
Because of the problems associated with
the construction (see page~\pageref{problem})
```

anywhere in the document, \LaTeX produces

```
┌
└ Because of the problems associated with the construction (see page 5)
└
```



Tip Beware, because of the way \LaTeX typesets a page, page references may be off by one.

See the discussion in Section 19.5 on how to guarantee that the page number is correct.

10.4.3 Floating tables and illustrations

Many documents contain tables and illustrations. These must be treated in a special way since they cannot be broken across pages. If necessary, \LaTeX moves—floats—a table or an illustration to the top or bottom of the current or the next page, if possible, and further away if not.

\LaTeX provides the `figure` and `table` environments for typesetting floats. The two are the same except that the `figure` environments are named Figure 1, Figure 2, and so on, whereas, the `table` environments are numbered as Table 1, Table 2, and so on.

Figures

Illustrations, also called *graphics* or *figures*, include drawings, scanned images, digitized photos, and so on. These can be inserted with a `figure` environment:

```
\begin{figure}
  Place the graphics here
  \caption{title}\label{Fi:xxx}
\end{figure}
```



Rule: The `\label` command must follow the `\caption` command.

Violation of this rule has unexpected consequences.

The above discussion of captions and labels for tables also applies to figures. If your document uses the `twocolumn` document class option, the `figure` environment produces figures that span only one column, but the `figure*` environment produces figures that span both columns. However, these figures can be placed only at the top of a page.



Tip Type

```
\begin{table}
  Place the table here
  \caption{title}\label{Ta:xxx}%Table~\ref{Ta:xxx}
\end{table}
```

to make cross-referencing quicker.

The standard way of including a graphics file is with the `\includegraphics` command provided by the `graphicx` package by David Carlisle and Sebastian Rahtz. Save your graphics in PDF format—as a rule.

Your graphics can also be made within a `picture` environment, an approach that is neither encouraged nor discussed in this book. To draw within \LaTeX , you may use `TikZ`; see Chapter 14.

Using the `graphicx` package, a typical figure is specified as follows:

```
\begin{figure}
  \centering\includegraphics{file}
  \caption{title}\label{Fi:xxx}
\end{figure}
```

The illustration `circle.pdf` is included with the command

```
\includegraphics{circle}
```

without the extension! \LaTeX and the `graphicx` package assumes the `pdf` extension.

If you have to scale the graphics image, say to 68% of its original size, use the command

```
\includegraphics[scale=.68]{file}
```

For instance, the figure on page 34 is included with the commands

```
\begin{figure}
  \centering\includegraphics[scale=.8]{StrucLaT}
  \caption{The structure of \protect\LaTeX.}
  \label{Fi:StrucLaT}
\end{figure}
```

For another use of the `graphicx` package, see Section 8.2.

Tables

We set up a table environment:

```
\begin{table}
  Place the table here
  \caption{title}\label{Ta:xxx}
\end{table}
```

A table environment can have more than one table, each with its own caption.

The `\caption` command is optional and may also precede the table. The optional `\label` command must be placed between the command `\caption` and the command `\end{table}`. The label is used to reference the table's number.



Tip Type

```
\begin{table}
  Place the table here
  \caption{title}\label{Ta:xxx}
\end{table}
```

to make cross-referencing quicker.

The `table` environment is primarily used for tables made with the `tabular` or similar environments (see Section 4.6). There are many examples of tables in this book; for instance, Section 3.4 has four. All the tables in this book were designed with the `booktabs` package, which is highly recommended.

If your document uses the `twocolumn` document class option, the `table` environment produces tables that span only one column and the `table*` environment produces tables that span both columns. Such tables can be placed only at the top of a page.

Float control

The `figure` and `table` environments may have an optional argument, with which you can influence \LaTeX 's placement of the typeset figure or table. The optional argument consists of one to four letters:

- `b`, the bottom of the page,
- `h`, here (where the environment appears in the text),
- `t`, the top of the page,
- `p`, a separate page.

For instance,

```
\begin{figure}[ht]
```

requests \LaTeX to place the figure “here” or at the “top” of a page. The default is `[tbh]` and the order of the optional arguments is immaterial, for example, `[th]` is the same as `[ht]`. If `h` is specified, it takes precedence, followed by `t` and `b`. The rules are complicated. LC3 dedicates more than 60 pages to this topic.

\LaTeX has about twenty internal parameters that control a complicated algorithm to determine the placement of tables and figures. To override these *for one table or figure only*, add an exclamation mark (!) to the optional argument. For instance, `[!h]` requests that this table or figure be placed where it is in the source file even if this placement violates the algorithm. For a detailed discussion of the float mechanism; see Chapter 7 of LC3.

The `\suppressfloats` command stops \LaTeX from placing any more tables or figures on the page it appears on. An optional argument `t` or `b` (but not both) prohibits placement of floats at the top or bottom of the current page. The table or figure that is *suppressed* appears on the next page or later in the document, if necessary.

Your demands and \LaTeX 's float mechanism may conflict with one another with the result that \LaTeX may not place material where you want it. The default values of the float placement parameters are good only for documents with a small number of floating objects.

There are two easy tools to use to make sure that the illustrations and tables are well-placed. Let us call the first:

Merging illustrations and tables.

Figure 10.2 is an example from my recent book [21]. In this book, there are three figures, as shown in Figure 10.2.

\LaTeX places the first on a page, and two, spread out, to the end of the chapter. So we recode the figures:

```
\begin{figure}[p]
\centerline{\includegraphics[scale=.9]{sc8}}
```

```

\caption{Defining gluing}\label{F:gluedef}

\bigskip
\bigskip

\centerline{\includegraphics[scale=.9]{sc9}}
\caption{An easy gluing example}\label{F:gluexamples}

\bigskip
\bigskip

\centerline{\includegraphics[scale=.9]{sc10}}
\caption{The two characteristic nondistributive lattices}
\label{F:n5m3}
\end{figure}

```

to get page 28 of [21]; see Figure 10.2. Note the use of the command `\bigskip` to keep the PDF files apart.

And the second tool is

The `\clearpage` command.

It not only starts a new page as the `\newpage` command, but also forces \LaTeX to print all the figures and tables it has accumulated but not yet placed in the typeset document. See also some related commands discussed in Section 3.6.3.

For more information on graphics; see Chapter 7 of LC3 and Chapter 2 of *The \LaTeX Graphics Companion* [12].

10.5 Back matter

The back matter of an article is very simple, as a rule. It is either empty or the bibliography. A long document, such as a book, may have more complicated back matter (see Chapter 19). In this section, we discuss only the *bibliography* and a very simple *index*.

10.5.1 Bibliographies in articles

The simplest way to typeset a bibliography is to type it directly into the article. For an example, see the bibliography in the `art2.tex` article (on page 302).

The following bibliography contains two examples, one short and one long, of each of the seven most frequently used type of items.

You type the text of a bibliography in a `thebibliography` environment, as shown in the following examples.

```

\begin{thebibliography}{99}
\bibitem{hA70}

```

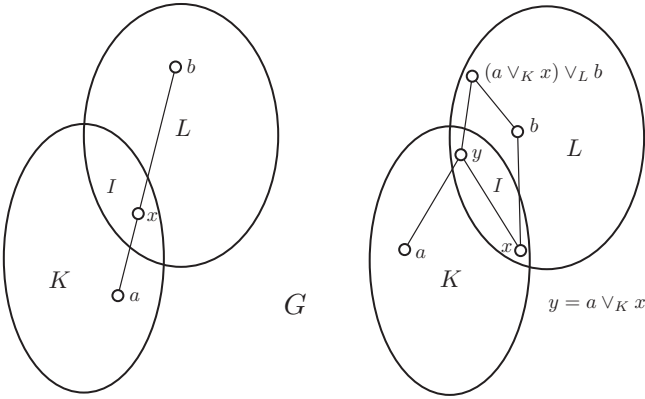


Figure 2.8: Defining gluing

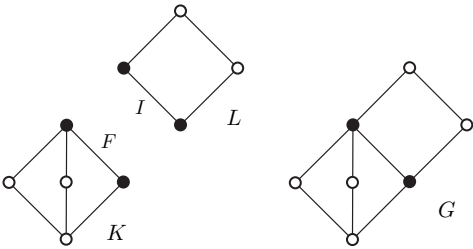


Figure 2.9: An easy gluing example

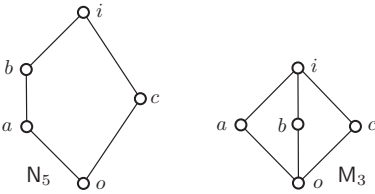


Figure 2.10: The two characteristic nondistributive lattices

Figure 10.2: Page 28 of my book [21]

Henry~H. Albert,
 Free torsoids,
 Current Trends in Lattice Theory.
 D.~Van Nostrand, 1970.

\bibitem{hA70a}
 Henry~H. Albert,
 Free torsoids.
 Current Trends in Lattice Theory
 (G.\,H. Birnbaum, ed.).
 vol.~7, D.~Van Nostrand, Princeton, January, 1970,
 no translation available, pp.~173--215 (German).

\bibitem{sF90}
 Soo-Key Foo,
 Lattice Constructions.
 Ph.D. thesis, University of Winnebago, 1990.

\bibitem{sF90a}
 Soo-Key Foo,
 Lattice Constructions.
 Ph.D. thesis, University of Winnebago, Winnebago, MN,
 December 1990, final revision not yet available.

\bibitem{gF86}
 Grant~H. Foster,
 Computational complexity in lattice theory,
 Tech. report, Carnegie Mellon University, 1986.

\bibitem{gF86a}
 Grant~H. Foster,
 Computational complexity in lattice theory.
 Research Note 128A, Carnegie Mellon University,
 Pittsburgh, PA, December, 1986,
 research article in preparation.

\bibitem{pK69}
 Peter Konig,
 Composition of functions.
 Proceedings of the Conference on Universal Algebra
 (Kingston, 1969).

\bibitem{pK69a}
 Peter Konig,
 Composition of functions.
 Proceedings of the Conference on Universal Algebra
 (G.~H. Birnbaum, ed.).
 vol.~7, Canadian Mathematical Society,
 Queen's Univ., Kingston, ON,

```

    available from the Montreal office,
    pp.~1--106 (English).
\bbitem{wL75}
    William~A. Landau,
    Representations of complete lattices.
    Abstract: Notices Amer. Math. Soc. \textbf{18}, 937.
\bbitem{wL75a}
    William~A. Landau,
    Representations of complete lattices.
    Abstract: Notices Amer. Math. Soc. \textbf{18}, 937,
    December, 1975.
\bbitem{gM68}
    George~A. Menuhin,
    Universal algebra.
    D.~Van Nostrand, Princeton, 1968.
\bbitem{gM68a}
    George~A. Menuhin,
    Universal algebra. 2nd ed.,
    University Series in Higher Mathematics, vol.~58,
    D.~Van Nostrand, Princeton,
    March, 1968 (English), no Russian translation.
\bbitem{eM57}
    Ernest~T. Moynahan,
    On a problem of M. Stone.
    Acta Math. Acad. Sci. Hungar.
    \textbf{8}~(1957), 455--460.
\bbitem{eM57a}
    Ernest~T. Moynahan,
    On a problem of M. Stone.
    Acta Math. Acad. Sci. Hungar.
    \textbf{8}~(1957), 455--460
    (English), Russian translation available.
\end{thebibliography}

```

Figure 10.3 shows a typeset version of this bibliography in the `amsart` document class.

You can find these entries in the document `AMSrefs.tex` in the `samples` folder (see page 5).

I use the convention that the label for a `\bbitem` consists of the initials of the author and the year of publication. The first cited publication by Andrew B. Reich in 1987 would have the label `aR87` and the second, `aR87a`. Of course, you can use any label you choose, but such conventions make the items easier to reuse.

The `thebibliography` environment takes an argument—in the previous example, it is 99—telling \LaTeX that the widest reference number it must generate is two digits wide. For fewer than 10 items, use 9 and for 100 or more items, use 999.

If the argument of `\begin{thebibliography}` is missing, you get the message

```
! LaTeX Error: Something's wrong--perhaps
    a missing \item.
```

Each bibliographic item is introduced with `\bibitem`, which performs the same function as the `\label` command for references. In your text, use `\cite`, in a similar way to `\eqref`—it provides the number enclosed in brackets. So if the 13th bibliographic item is introduced with

```
\bibitem{eM57}
```

then

```
\cite{eM57}
```

REFERENCES

- [1] Henry H. Albert, Free toroids, Current Trends in Lattice Theory. D. Van Nostrand, 1970.
- [2] Henry H. Albert, Free toroids. Current Trends in Lattice Theory (G.H. Birnbaum, ed.). vol. 7, D. Van Nostrand, Princeton, January, 1970, no translation available, pp. 173–215 (German).
- [3] Soo-Key Foo, Lattice Constructions. Ph.D. thesis, University of Winnebago, 1990.
- [4] Soo-Key Foo, Lattice Constructions. Ph.D. thesis, University of Winnebago, Winnebago, MN, December 1990, final revision not yet available.
- [5] Grant H. Foster, Computational complexity in lattice theory, Tech. report, Carnegie Mellon University, 1986.
- [6] Grant H. Foster, Computational complexity in lattice theory. Research Note 128A, Carnegie Mellon University, Pittsburgh, PA, December, 1986, research article in preparation.
- [7] Peter König, Composition of functions. Proceedings of the Conference on Universal Algebra (Kingston, 1969).
- [8] Peter König, Composition of functions. Proceedings of the Conference on Universal Algebra (G. H. Birnbaum, ed.). vol. 7, Canadian Mathematical Society, Queen's Univ., Kingston, ON, available from the Montreal office, pp. 1–106 (English).
- [9] William A. Landau, Representations of complete lattices. Abstract: Notices Amer. Math. Soc. **18**, 937.
- [10] William A. Landau, Representations of complete lattices. Abstract: Notices Amer. Math. Soc. **18**, 937, December, 1975.
- [11] George A. Menuhin, Universal algebra. D. Van Nostrand, Princeton, 1968.
- [12] George A. Menuhin, Universal algebra. 2nd ed., University Series in Higher Mathematics, vol. 58, D. Van Nostrand, Princeton, March, 1968 (English), no Russian translation.
- [13] Ernest T. Moynahan, On a problem of M. Stone. Acta Math. Acad. Sci. Hungar. **8** (1957), 455–460.
- [14] Ernest T. Moynahan, On a problem of M. Stone. Acta Math. Acad. Sci. Hungar. **8** (1957), 455–460 (English), Russian translation available.

Figure 10.3: The most important bibliographic entry types.

refers to that item and typesets it as [13]. The bibliography of the article itself is automatically numbered by \LaTeX . It is up to the author to make sure that the listing of the bibliographic items is in the proper order. (Note that `bibtex` takes care of the ordering.)



Tip Do not leave spaces in a `\cite` command. For example, `\cite{eM57 }` produces [?] indicating an unknown reference.

You can use `\cite` to cite two or more items in the form

```
\cite{hA70,eM57}
```

which typesets as [1, 13]. There is also an optional argument for `\cite` to specify additional information. For example,

```
\cite[pages~2--15]{eM57}
```

typesets as [13, pages 2–15].

If you wish to use labels rather than numbers to identify bibliographic items, then you can specify those labels with an optional argument of the `\bibitem` command:

```
[
[EM57] Ernest T. Moynahan, On a problem of M. Stone, Acta Math. Acad. Sci. Hungar. 8 (1957),
455–460.
]
```

typed as

```
\bibitem[EM57]{eM57}
Ernest~T. Moynahan, \emph{On a problem of M. Stone},
Acta Math. Acad. Sci. Hungar.
\textbf{8} (1957), 455--460.
```

If this optional argument of `\bibitem` is used, then the `\cite` command produces [EM57]. The argument of `\begin{thebibliography}` must be set wide enough to allow for such labels.



Rule: Label for a bibliographic item

A label cannot contain a comma or a space.

The examples I have used follow the formatting rules set by the AMS. Titles are not italicized, and only volume numbers of journals are set in boldface. You also have to watch the order in which the items are given, the punctuation, and the capitalization.

If an author appears repeatedly, use the `\bysame` command, which replaces the author's name with a long dash followed by a thin space. For example,

```
\bibitem{gF86}
  Grant~H. Foster,
  \emph{Computational complexity in lattice theory},
  tech. report, Carnegie Mellon University, 1986.
\bibitem{gF86a}
  \bysame,
  \emph{Computational complexity in lattice theory},
  Research Note 128A, Carnegie Mellon University,
  Pittsburgh PA, December 1986,
  research article in preparation.
```

See the third page of `art2.pdf` on page [302](#) for a typeset example.



Tip If you want a different title for your bibliography, say `Bibliography`, place the command

```
\renewcommand{\refname}{Bibliography}
```

anywhere before the `thebibliography` environment (see Section [15.1.7](#)). If you use a legacy document class or `amsbook.cls`, use the line

```
\renewcommand{\bibname}{Bibliography}
```



Tip You may have more than one `thebibliography` environment in a document. Because each bibliography would number the entries from 1, you should provide labels as optional parameters of the `\bibitem` commands for cross-referencing.

10.5.2 Simple indexes

Using the `\label` and `\pageref` commands (see Section [10.4.2](#)), it is quite simple to produce a small index in a `theindex` environment. At each point in the text that you want to reference in the index, place a `\label` command. The corresponding entry in the index typesets the page number with the `\pageref` command.

The `\item`, `\subitem`, and `\subsubitem` commands create an entry, subentry, and subsubentry, respectively. If you need additional vertical spacing when the first letter changes, for instance, between the “h” entries and the “i” entries, you can use the

`\indexspace` command. Here are some examples of index entries:

```
\begin{theindex}
\item Lakser, H., \pageref{Lakser}
\item Lattice, \pageref{Lattice_intro},
      \textbf{\pageref{Lattice}}
      \subitem distributive, \pageref{Lattice_distributive}
      \subitem modular, \pageref{Lattice_distributive},
        \textbf{\pageref{Lattice_distributive2}}
\item Linear subspace, \pageref{Linear_subspace}
\end{theindex}
```

And here is the typeset index:

INDEX

Lakser, H., 2
 Lattice, 14, **25**
 distributive, 18
 modular, 19, **37**
 Linear subspace, 38

For a larger index, you should use the *MakeIndex* application (see Chapter 18).

10.6 Visual design

In this chapter, we have discussed the logical design of a \LaTeX document. The visual design is largely left to the document class. But there is one small aspect of the visual design we have to discuss, the page style.

To get a visual representation of the page style of your document, use `layout` package of Kent McPherson. Load the package with

```
\usepackage{layout}
```

and place the `\layout` command somewhere in the body of your article. \LaTeX produces a graphical representation of the page layout. Figure 10.4 shows the page layout for odd pages for the `amsart` document class with no options.

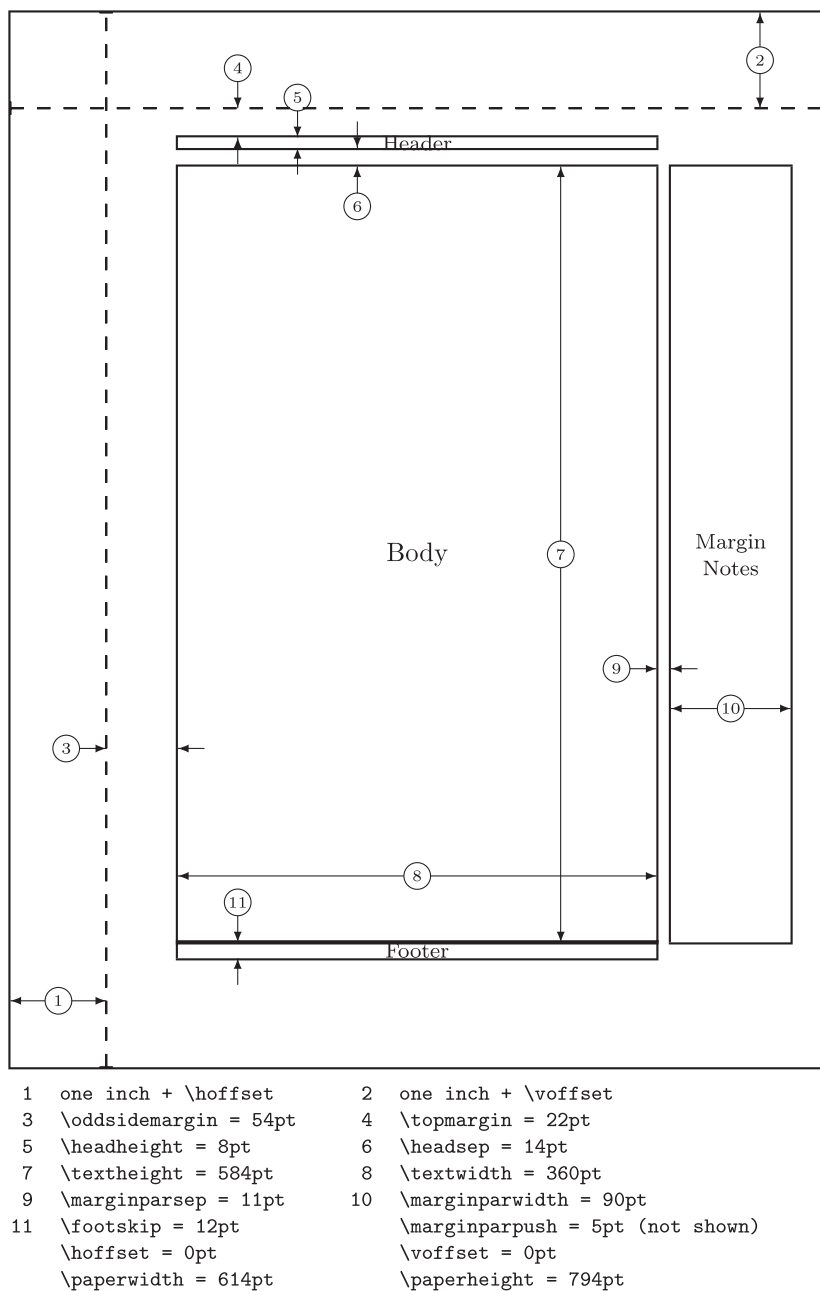
A typeset page has three parts, the *running head* or *header*, the *body*, and the *footer*. As a rule, the document class takes care of the contents and formatting of all three parts. For the running head and footer, however, you can override the page design of the document class with the command

```
\pagestyle{style}
```

where the argument *style* is one of the following:

plain The running head is empty and the footer contains only the page number.

empty Both the running head and the footer are blank.

Figure 10.4: Page layout for the `amsart` document class.

`headings` The running head contains the information provided by the document class and the footer is empty.

`myheadings` The running head contains the information provided by the commands `\markboth` and `\markright`, the footer is empty.

The `\markright` command takes only one argument. The last `\markright` on a page provides the running head information for that page. The `\markboth` command has two arguments. The first provides the running head information for a left-hand page, the second provides the running head information for a right-hand page. The AMS document classes also have a `\markleft` command for the running head information for a left-hand page.

The `\thispagestyle` command is the same as `\pagestyle` except that it affects only the current page.

For instance, if the current page is a full-page graphic, you might want to issue the command

```
\thispagestyle{empty}
```

The `\maketitle` command automatically issues a

```
\thispagestyle{plain}
```

command, so if you want to suppress the page number on the first page of a document, you have to put

```
\thispagestyle{empty}
```

immediately after the `\maketitle` command.

The commands listed in Figure 10.4 are length commands (see Section 15.5.2) and can be changed with the commands introduced in that section. As a rule, you do not have to worry about these settings, they are chosen by the document class for you. Sometimes, however, you have a job that requires such changes. I once had to submit a research plan on a form with a 7.5 inch by 5 inch box. To be able to cut and paste the typeset report, I had to produce the text with a `\textwidth` of 7 inches. If I simply set

```
\setlength{\textwidth}{7in}
```

the text would overflow the printed page and the last few characters of each line would be missing. So I had to change the margins by starting the document with

```
\documentclass[12pt]{report}
\setlength{\textwidth}{7in}
\setlength{\oddsidemargin}{2pt}
```

All of Chapter 5 of LC3 deals with page layouts. There you can find a description of the `geometry` package of Hideo Umeki, which computes all the parameters from

the ones you supply. Also you find there a discussion of Piet van Oostrum's excellent package, `fancyhdr`, which allows you to create your own page style (see also [16]).

However, if you submit an article to a journal, do not change the type size, page dimensions, headers. Use the document class and the article templates the journal provides (if any). This will make your submission easier for you and the journal. See also Section 2.6.



The AMS article document class

In this chapter, we discuss `amsart`, the main AMS document class for journal articles. The AMS book document class is discussed in Chapter 19.

In Section 11.1, I argue that there are good reasons why you should write your articles for publication in `amsart`. Section 11.2 introduces the rules governing the top matter in the `amsart` document class. The `amsart` sample article `art2.tex` is presented in Section 11.3. In Section 11.4, you are guided through the process for creating detailed templates.

A document class is finely tuned by its options. In Section 11.5, we discuss the options of `amsart`. Section 11.6 briefly describes the various packages in the AMS distribution and their interdependencies.

11.1 *Why amsart?*

11.1.1 *Submitting an article to the AMS*

You want to submit an article written with the `amsart` document class to the Proceedings of the American Mathematical Society.

For general information on the AMS journals, go to the AMS Web site

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-55281-6_11

<http://www.ams.org/arc/>

and in the Author Resource Center start discovering a wealth of information.

Now, in the preamble of your article, replace the line

```
\documentclass{amsart}
```

with

```
\documentclass{proc-l}
```

Typeset the article and you are done. Your article is formatted as it will appear in the Proceedings.

11.1.2 Submitting an article to *Algebra Universalis*

There are many journals whose document classes are based on `amsart`. For instance,

<http://www.algebrauniversalis.com>

takes you to the home page of the journal *Algebra Universalis*.

To find the document class, click on `Instructions for Authors` and in Section C, click on `AUfiles.zip`, which provides the files you need. Now in your article make the replacement:

```
\documentclass{birkau}
```

and your article typesets in the format appropriate for this journal.

11.1.3 Submitting to other journals

A large number of journals use document classes based on `amsart`. Not all are as friendly as *Algebra Universalis*, but as a rule a small number of changes in the article suffice.

All of them share the attribute that the top matter is given as the arguments of several commands. In the introductory sample article, `art1.tex`, on page 25, there was only one, but in the sample article `art2.tex` in Section 11.3, there are quite a few—there could be more. As a result, this document class is able to shape the top matter as the journal requires. Even if the names of some of these commands are different (e.g., `affiliation` for `address`), the principles you learn from the `amsart` document class apply.

Many journals insist that you use their own document classes. For these, you may have to add the AMS packages (see Section 11.6) to continue using the enhancements of the AMS.

A shrinking number of journals use document classes incompatible with the AMS packages. Was kann man machen?

11.1.4 Submitting to conference proceedings

The AMS also has a document class for articles in book-form proceedings of meetings. The differences in the rules for the `amsart` and `amsproc` document classes are minor; for instance, `amsproc` does not access `\date`.

11.2 The top matter

See the typeset top matter of the `art2.tex` article on pages 299 and 302 for a fairly representative example. As you may recall from Section 1.8, part of the author information is moved to the end of the typeset article—see page 302.

Title page information is provided as arguments of several commands. For your convenience, I divide them into three groups: information about the article, information about the author, and AMS -related information.

There is only one general rule.



Rule: Top matter commands

All top matter commands are *short*.

This means that there can be no blank line (or `\par` command) in the argument of any of these commands (see Section 3.3.3).

11.2.1 Article information

You have to supply five pieces of information about the article.



Rule: Title

- Command: `\title`.
 - Separate lines with `\\`.
 - Optional argument: Short title for running head.
 - Do not put a period at the end of a title.
 - Do not use custom commands in the title.
-

The typeset title is placed on the front page of the typeset article.

Many titles are too long to be typeset on a single line. If the way \LaTeX breaks the title is not satisfactory, you can indicate where the title should be broken with the

`\` command. Alternatively, you may nudge L^AT_EX in the right direction with `~` (see Section 3.4.3). For instance, the title:

The `\texttt{amsart}` document class

is broken by L^AT_EX between `document` and `class`. So either add `\`:

The `\texttt{amsart}\` document class

or replace `document class` with `document~class`:

The `\texttt{amsart}` document~class

The *running head* (see Section 10.6) is the title on odd-numbered pages, set in capital letters. If the title is more than a few words long, use an optional argument to specify a short title for the running head. Do not use `\` in the short title.

Example of a title:

`\title{A construction of distributive lattices}`

A title with a short title:

`\title[Complete-simple distributive lattices]`
`{A construction of complete-simple`
`distributive lattices}`

Note the AMS rules about short titles and the table of contents in Section 19.2.1.



Rule: Translator

- Command: `\translator`.
- Do not put a period at the end of the argument.

The typeset `\translator` is placed on the last page of the typeset article, before the address(es). There can be more than one translator. Each should be given as the argument of a separate `\translator` command. *Example:*

`\translator{Harry~M. Goldstein}`



Rule: Dedication

- Command: `\dedicatory`.
 - Separate lines with `\`.
-

The typeset dedication is placed under the author(s).

Example:

```
\dedicatory{To the memory of my esteemed
friend and teacher,\ Harry~M. Goldstein}
```



Rule: Date

- Command: `\date`.

Example:

```
\date{January 22, 2015}
```

The typeset `\date` is placed on the front page of the typeset article as a footnote. Do not use `\date` when you submit an article; specify the submission date.

To suppress the date, use `\date{}` or omit the `\date` command entirely.

11.2.2 Author information

There are seven pieces of information about yourself.



Rule: Author

- Command: `\author`.
- Optional argument: Short form of the name for the running head.

The typeset `author` is placed on the front page of the typeset article.

Examples:

```
\author{George~A. Menuhin}
```

With a short form of the name for the running head:

```
\author[G.\,A. Menuhin]{George~A. Menuhin}
```

Section [11.2.4](#) discusses how to specify multiple authors.

**Rule: Contributor**¹

- Command: `\contrib`.
- Optional argument: Describing the contribution.

The typeset contributor's name is placed on the front page of the typeset article.

Examples:

A contributor authoring an appendix:

```
\contrib[with an appendix by]{John Blaise}
```

If this appendix has two authors:

```
\contrib[with an appendix by]{J. Blaise}
\contrib[] {W. Brock}
```

This typesets (with author G. A. Menuhin) the author line as

```
[
G. A. MENUHIN, WITH AN APPENDIX BY J. BLAISE AND W. Brock
]
```

Contributors can have addresses, current addresses, etc., just like authors.

**Rule: Address**

- Command: `\address`.
- Separate lines with `\\`.
- Optional argument: Name of author.

The typeset address is placed at the end of the typeset article.

Example:

```
[
DEPARTMENT OF APPLIED MATHEMATICS, UNIVERSITY OF WINNEBAGO, WINNEBAGO, MN 53714
]
```

which is typed as

```
\address{Department of Applied Mathematics\\
          University of Winnebago\\
          Winnebago, MN 53714}
```

¹As I am writing this, the command does not work. This has been publicly discussed since July 2018.

Notice that L^AT_EX replaces the `\\` line separators with commas.

Notice that, with the `amsart` document class, LaTeX replaces... (it doesn't happen with `art`)

If there are several authors, you can use the author's name as an optional argument of `\address` to avoid ambiguity. See Example 4 in Section 11.2.5 (page 291) for a complete example.



Rule: Current address

- Command: `\curraddr`.
- Separate lines with `\\`.
- Optional argument: name of author.

The typeset current address is placed at the end of the typeset article.

Example:

┌
Current address: Department of Mathematics, University of York, Heslington, York,
 England
 └

is typed as

```
\curraddr{Department of Mathematics\\
          University of York\\
          Heslington, York, England}
```

If there are several authors, you can use the author's name as an optional argument of `\curraddr` to avoid ambiguity; for some examples; see Section 11.2.5.



Rule: E-mail address

- Command: `\email`.
- Optional argument: Name of author.

The typeset e-mail address is placed at the end of the typeset article.

Example:

```
\email{gmen@ccw.uwinnebago.edu}
```



Tip Some e-mail addresses contain the special underscore character (`_`). Recall (see Section 3.4.4) that you have to type `_` to get it.

Example:

```
\email{George\_Gratzer@umanitoba.ca}
```



Tip Some older e-mail addresses contain the percent symbol (`%`); recall that you have to type `\%` to get it (see Section 3.4.4).

Example:

```
\email{h1175moy\%ella@relay.eu.net}
```



Rule: Web (home) page (URL)

- Command: `\urladdr`.
 - Optional argument: Name of author.
-

The typeset Web (home) page is placed at the end of the typeset article.

Example:

```
\urladdr{http://www.maths.umanitoba.ca/homepages/gratzer/}
```



Tip Many Internet addresses contain the tilde (`~`), indicating the home directory of the user. Type `~` to get `~` and not `\~`; see Section 3.4.4. `\sim` is also unacceptable.

Example:

```
\urladdr{http://kahuna.math.hawaii.edu/~ralph/}
```

**Rule: Research support or other acknowledgments**

- Command: `\thanks`.
- Do not specify linebreaks.
- Terminate the sentence with a period.

The typeset research support or other acknowledgments is placed on the front page of the typeset article as an unmarked footnote.

Example:

```
\thanks{Supported in part by NSF grant PAL-90-2466.}
```

A `\thanks{}` command is ignored in typesetting.

11.2.3 AMS information

The AMS requires that you supply two more pieces of information about the article.

The following are collected at the bottom of the first page as unmarked footnotes along with the arguments of the `\thanks` and `\date` commands.

**Rule: AMS subject classifications**

- Command: `\subjclass`.
- Optional argument: 2020.
- The argument should be either a five-character code or the phrase `Primary:` followed by a five-character code, a semicolon, the phrase `Secondary:` and one or more additional five-character codes.

The typeset AMS subject classifications is placed at the bottom of the front page of the typeset article as a footnote.

Examples:

```
\subjclass[2020]{06B10}
```

```
\subjclass[2020]{Primary: 06B10; Secondary: 06D05}
```

The current subject classification scheme for mathematics was adopted in 2020, making the 1991 and 2010 classification schemes obsolete. Thus, 2020 should be considered as a *compulsory* optional argument—maybe the only one in all of \LaTeX . Maybe, in 2030 and after, it should be 2030.

The current subject classification scheme, MSC 2020, is available from the AMS Web site

<http://www.ams.org/>

Search for MSC. Or in the Author Resource Center click on MSC.



Rule: Keywords

- Command: `\keywords`.
- Do not indicate line breaks.
- `amsart` supplies the phrase *Key words and phrases*. and a period at the end of the list of keywords.

The typeset keywords are placed on the front page of the typeset article as a footnote on the first page.

Example:

```
\keywords{Complete lattice, distributive lattice,
complete congruence, congruence lattice}
```

Keywords are optional for many journals.

Further footnotes An additional `\thanks` command creates an unmarked footnote.

Examples:

```
\thanks{This is a preliminary version of this article,
        prepared for the Second Annual Meeting of the
        Statistical Association of Winnebago.}
```

```
\thanks{This article is in final form, and no version
        of it will be submitted elsewhere.}
```

11.2.4 Multiple authors

If an article has several authors, repeat the author information commands for each one. Take care that an e-mail address follows the address.

If two authors share the same address, omit the `\address` command for the second author, who can still have a different e-mail address and Web home page. An additional `\thanks` command for the first author should precede any `\thanks` commands for the second author. Since the footnotes are not marked, the argument of the `\thanks` command for research support should contain a reference to the author:

```
\thanks{The research of the first author was supported
        in part by NSF grant PAL-90-2466.}
```

```
\thanks{The research of the second author was supported by
        the Hungarian National Foundation for Scientific
        Research, under Grant No.~9901.}
```

Finally, if an article has more than two authors, supply the author information for each author as usual, but explicitly specify the running heads with the `\markleft` command:

```
\markleft{FIRST AUTHOR ET AL.}
```

where *FIRST AUTHOR* must be all capitals.

If there are multiple authors, sometimes it may not be clear whose address, current address, e-mail address, or Web home page is being given. In such cases, give the name of the authors as optional arguments for these commands. For example,

```
[
Email address, Ernest T. Moynahan: emoy@ccw.uwinnebago.edu.
]
```

is typed as

```
\email[Ernest~T. Moynahan]{emoy@ccw.uwinnebago.edu}
```

See also Example 4 in Section 11.2.5.

11.2.5 Examples

The following examples show typical top matter commands and can be found in the `topmat.tpl` file in the `samples` folder (see page 5).

Example 1 One author.

```
%Article information
\title[Complete-simple distributive lattices]
      {A construction of complete-simple\\
       distributive lattices}
\date{\today}

%Author information
\author{George~A. Menuhin}
\address{Computer Science Department\\
         University of Winnebago\\
         Winnebago, MN 53714}
\email{gmen@ccw.uwinnebago.edu}
```

```
\urladdr{http://math.uwinnebago.edu/homepages/menuhin/}
\thanks{This research was supported by
       the NSF under grant number 23466.}
```

```
%AMS information
\keywords{Complete lattice, distributive lattice,
          complete congruence, congruence lattice}
\subjclass[2010]{Primary: 06B10; Secondary: 06D05}
```

In the `\title` command, supplying the optional argument for the running head is the rule, not the exception. The only required item is `\title`. If it is missing, you get the strange message:

```
! Undefined control sequence.
<argument> \shorttitle
```

1.49 `\maketitle`

Example 2 Two authors but only the first has a Web home page. I only show the author information section here. The other commands are the same as in Example 1.

```
%Author information
\author{George~A. Menuhin}
\address{Computer Science Department\\
         University of Winnebago\\
         Winnebago, MN 53714}
\email{gmen@ccw.uwinnebago.edu}
\urladdr{http://math.uwinnebago.edu/homepages/menuhin/}
\thanks{The research of the first author was
       supported by the NSF under grant number 23466.}
\author{Ernest~T. Moynahan}
\address{MTA R\'enyi Institute of Mathematics\\
         Budapest, P.O.B. 127, H-1364\\
         Hungary}
\email{h1175moy\%ella@relay.eu.net}
\thanks{The research of the second author
       was supported by the Hungarian
       National Foundation for Scientific Research,
       under Grant No. 9901.}
```

Example 3 Two authors, same department. I only show the author information section here. The other commands are identical to those in Example 1.


```
%Author information
\author{George~A. Menuhin}
\address{Computer Science Department\\
         University of Winnebago\\
         Winnebago, MN 53714}
\email[George~A. Menuhin]{gmen@ccw.uwinnebago.edu}
\urladdr[George~A. Menuhin]{%
    {http://math.uwinnebago.edu/homepages/menuhin/}}
\thanks{The research of the first author was
        supported by the NSF under grant number~23466.}
\author{Ernest~T. Moynahan}
\email[Ernest~T. Moynahan]{emoy@ccw.uwinnebago.edu}
\thanks{The research of the second author was supported
        by the Hungarian National Foundation for
        Scientific Research, under Grant No. 9901.}
```

Note that the second author has no \address.

Example 4 Three authors, the first two from the same department, the second and third with e-mail addresses and research support. I only show the author information section. The other commands are unchanged. There are various ways of handling this situation. This example shows one solution.

```
%Author information
\author{George~A. Menuhin}
\address[George~A. Menuhin and Ernest~T. Moynahan]
    {Computer Science Department\\
     University of Winnebago\\
     Winnebago, MN 53714}
\email[George~A. Menuhin]{gmen@ccw.uwinnebago.edu}
\urladdr[George~A. Menuhin]{%
    {http://math.uwinnebago.edu/homepages/menuhin/}}
\thanks{The research of the first author was
        supported by the NSF under grant number 23466.}
\author{Ernest~T. Moynahan}
\email[Ernest~T. Moynahan]{emoy@ccw.uwinnebago.edu}
\thanks{The research of the second author was supported
        by the Hungarian National Foundation for
        Scientific Research, under Grant No. 9901.}
\author{Ferenc~R. Richardson}
\address[Ferenc~R. Richardson]
    {Department of Mathematics\\
     California United Colleges\\
     Frasco, CA 23714}
```

```
\email[Ferenc~R. Richardson]{frich@ccu.frasco.edu}
\thanks{The research of the third author was
        supported by the NSF under grant number 23466.}
```

The most common mistake in the top matter is the misspelling of a command name; for instance, `\adress`. L^AT_EX sends the error message

```
! Undefined control sequence.
1.37 \adress
        {Computer Science Department\\
```

which tells you exactly what you mistyped. Similarly, if you drop a closing brace, as in

```
\email{menuhin@ccw.uwinnebago.edu
```

you are told clearly what went wrong. Because the top matter commands are short (see Section 3.3.3), L^AT_EX gives the message

```
Runaway argument?
{menuhin@ccw.uwinnebago.edu \thanks
    {The research of th\ETC.
!File ended while scanning use of \\email.
```

If you drop an opening brace,

```
\author George~A. Menuhin}
```

you get the message

```
! Too many }'s.
1.43 \author George~A. Menuhin}
```

If you enclose an optional argument in braces instead of brackets,

```
\title{Complete-simple distributive lattices}%
        {A construction of complete-simple\\
        distributive lattices}
```

L^AT_EX uses the short title as the title and the real title is typeset before the title of the typeset article.

11.2.6 Abstract

As we discussed in Section 10.3, you type the abstract in an `abstract` environment, which you place as the last item before the `\maketitle` command. The abstract should be self-contained; do not include cross-references and do not cite from the bibliography. Avoid custom commands.

If you place the abstract *after* the `\maketitle` command, L^AT_EX typesets it wherever it happens to be and sends a warning.

11.3 The sample article

art2.tex is the source file for our more advanced sample article (in the samples folder, see page 5) using the amsart document class. A simpler article, art1.tex, is presented in Part I (see Section 1.8).

Here is art2.tex followed by the typeset form.

```
% Sample file: art2.tex
\documentclass{amsart}
\usepackage{amssymb,latexsym}

\theoremstyle{plain}
\newtheorem{theorem}{Theorem}
\newtheorem{corollary}{Corollary}
\newtheorem*{main}{Main~Theorem}
\newtheorem{lemma}{Lemma}
\newtheorem{proposition}{Proposition}

\theoremstyle{definition}
\newtheorem{definition}{Definition}
\theoremstyle{remark}
\newtheorem*{notation}{Notation}

\numberwithin{equation}{section}

\begin{document}
\title[Complete-simple distributive lattices]
      {A construction of complete-simple\\
       distributive lattices}
\author{George~A. Menuhin}
\address{Computer Science Department\\
         University of Winnebago\\
         Winnebago, MN 53714}
\email{menuhin@ccw.uwinnebago.edu}
\urladdr{http://math.uwinnebago.edu/homepages/menuhin/}
\thanks{Research supported by the NSF under grant number
23466.}
\keywords{Complete lattice, distributive lattice,
          complete congruence, congruence lattice}
\subjclass[2020]{Primary: 06B10; Secondary: 06D05}
\date{March 15, 2024}
\begin{abstract}
  In this note we prove that there exist
```

```

\emph{complete-simple distributive lattices,}
that is, complete distributive lattices in which
there are only two complete congruences.
\end{abstract}

\maketitle

\section{Introduction}\label{S:intro}
In this note we prove the following result:

\begin{main}
  There exists an infinite complete distributive lattice~ $K$ 
  with only the two trivial complete congruence relations.
\end{main}

\section{The  $D^{\{\langle 2 \rangle\}}$  construction}\label{S:Ds}
For the basic notation in lattice theory and universal algebra,
see Ferenc~R. Richardson~\cite{fR82}
and George~A. Menuhin~\cite{gM68}.
We start with some definitions:

\begin{definition}\label{D:prime}
  Let  $V$  be a complete lattice, and let  $\mathfrak{p} = [u, v]$ 
  be an interval of  $V$ . Then  $\mathfrak{p}$  is called
  \emph{complete-prime} if the following
  three conditions are satisfied:
  \begin{enumerate}
    \item  $u$  is meet-irreducible but  $u$  is \emph{not}
      completely meet-irreducible;\label{m-i}
    \item  $v$  is join-irreducible but  $v$  is \emph{not}
      completely join-irreducible;\label{j-i}
    \item  $[u, v]$  is a complete-simple lattice.\label{c-s}
  \end{enumerate}
\end{definition}

Now we prove the following result:

\begin{lemma}\label{L:ds}
  Let  $D$  be a complete distributive lattice satisfying
  conditions \eqref{m-i} and \eqref{j-i}. Then
   $D^{\{\langle 2 \rangle\}}$  is a sublattice of  $D^{\{2\}}$ ;
  hence  $D^{\{\langle 2 \rangle\}}$  is a lattice, and

```

$D^{\langle 2 \rangle}$ is a complete distributive lattice satisfying conditions \eqref{m-i} and \eqref{j-i}.

\begin{proof}

By conditions \eqref{m-i} and \eqref{j-i}, $D^{\langle 2 \rangle}$ is a sublattice of D^2 . Hence, $D^{\langle 2 \rangle}$ is a lattice.

Since $D^{\langle 2 \rangle}$ is a sublattice of a distributive lattice, $D^{\langle 2 \rangle}$ is a distributive lattice. Using the characterization of standard ideals in Ernest~T. Moynahan~\cite{eM57}, $D^{\langle 2 \rangle}$ has a zero and a unit element, namely, $\langle 0, 0 \rangle$ and $\langle 1, 1 \rangle$. To show that $D^{\langle 2 \rangle}$ is complete, let $\varnothing \neq A \subseteq D^{\langle 2 \rangle}$, and let $a = \bigvee A$ in D^2 . If $a \in D^{\langle 2 \rangle}$, then $a = \bigvee A$ in $D^{\langle 2 \rangle}$; otherwise, a is of the form $\langle b, 1 \rangle$ for some $b \in D$ with $b < 1$. Now $\bigvee A = \langle 1, 1 \rangle$ in D^2 and the dual argument shows that $\bigwedge A$ also exists in D^2 . Hence D is complete. Conditions \eqref{m-i} and \eqref{j-i} are obvious for $D^{\langle 2 \rangle}$.

\end{proof}

$\begin{corollary}\label{C:prime}$

If D is complete-prime, then so is $D^{\langle 2 \rangle}$.

$\end{corollary}$

The motivation for the following result comes from Soo-Key Foo~\cite{sF90}.

$\begin{lemma}\label{L:ccr}$

Let Θ be a complete congruence relation of $D^{\langle 2 \rangle}$ such that

$\begin{equation}\label{E:rigid}$

$\langle 1, d \rangle \equiv \langle 1, 1 \rangle \pmod{\Theta},$

$\end{equation}$

for some $d \in D$ with $d < 1$. Then $\Theta = \iota$.

\end{lemma}

`\begin{proof}`

Let Θ be a complete congruence relation of $D^{\angle 2 \angle}$ satisfying [\eqref{E:rigid}](#).

Then $\Theta = \iota$.

`\end{proof}`

`\section{The Π^* construction}\label{S:P*}`

The following construction is crucial to our proof of the Main Theorem:

`\begin{definition}\label{D:P*}`

Let D_i , for $i \in I$, be complete distributive lattices satisfying condition [\eqref{j-i}](#).

Their Π^* product is defined as follows:

`\[`

$$\Pi^* (D_i \mid i \in I) \\ = \Pi (D_i^{-} \mid i \in I) + 1;$$

`\]`

that is, $\Pi^* (D_i \mid i \in I)$ is

$\Pi (D_i^{-} \mid i \in I)$ with a new unit element.

`\end{definition}`

`\begin{notation}`

If $i \in I$ and $d \in D_i^{-}$, then

`\[`

$$\angle \dots, 0, \dots, \overset{i}{d}, \dots, 0, \\ \dots \angle$$

`\]`

is the element of $\Pi^* (D_i \mid i \in I)$ whose i -th component is d and all the other components are 0 .

`\end{notation}`

See also Ernest T. Moynahan [\cite{eM57a}](#). Next we verify:

`\begin{theorem}\label{T:P*}`

Let D_i , for $i \in I$, be complete distributive lattices satisfying condition [\eqref{j-i}](#).

Let Θ be a complete congruence

relation on $\Pi^* (D_i \mid i \in I)$. If there exist

$i \in I$ and $d \in D_i$ with $d < 1_i$ such that for

all $d \leq c < 1_i$,

```

\begin{equation}\label{E:cong1}
\langle \dots, 0, \dots, \overset{i}{d},
\dots, 0, \dots \rangle \equiv \langle \dots, 0, \dots,
\overset{i}{c}, \dots, 0, \dots \rangle \pmod{\Theta},
\end{equation}
then  $\Theta = \iota$ .
\end{theorem}

\begin{proof}
Since
\begin{equation}\label{E:cong2}
\langle \dots, 0, \dots, \overset{i}{d}, \dots, 0,
\dots \rangle \equiv \langle \dots, 0, \dots,
\overset{i}{c}, \dots, 0, \dots \rangle \pmod{\Theta},
\end{equation}
and  $\Theta$  is a complete congruence relation, it follows
from condition~\eqref{c-s} that
\begin{equation}\label{E:cong}
\begin{split}
&\langle \dots, \overset{i}{d}, \dots, 0, \\
&\quad \dots \rangle \\
&\equiv \bigvee ( \langle \dots, 0, \dots, \\
&\quad \overset{i}{c}, \dots, 0, \dots \rangle \mid d \leq c < 1 ) \\
&\equiv 1 \pmod{\Theta}.
\end{split}
\end{equation}

Let  $j \in I$  for  $j \neq i$ , and let  $a \in D_j^{-}$ .
Meeting both sides of the congruence \eqref{E:cong2} with
 $\langle \dots, 0, \dots, \overset{j}{a}, \dots, 0, \dots \rangle$ ,
we obtain
\begin{equation}\label{E:comp}
\begin{split}
0 &\equiv \langle \dots, 0, \dots, \overset{i}{d}, \dots, 0, \dots \\
&\quad \rangle \wedge \langle \dots, 0, \dots, \overset{j}{a}, \\
&\quad \dots, 0, \dots \rangle \\
&\equiv \langle \dots, 0, \dots, \overset{j}{a}, \dots, 0, \\
&\quad \dots \rangle \pmod{\Theta}.
\end{split}
\end{equation}
Using the completeness of  $\Theta$  and \eqref{E:comp}, we get:
\l[

```

```

0 \equiv \bigvee ( \langle \dots, 0, \dots, \overset{j}{a},
\dots, 0, \dots \rangle \mid a \in D_{j^{-}} )
= 1 \pmod{\Theta},
\]
hence  $\Theta = \iota$ .
\end{proof}

\begin{theorem}\label{T:P*a}
Let  $D_i$  for  $i \in I$  be complete distributive lattices
satisfying conditions \eqref{j-i} and~\eqref{c-s}. Then
 $\Pi^* ( D_i \mid i \in I )$  also satisfies
conditions~\eqref{j-i} and \eqref{c-s}.
\end{theorem}

\begin{proof}
Let  $\Theta$  be a complete congruence on
 $\Pi^* ( D_i \mid i \in I )$ . Let  $i \in I$ . Define
\[
\widehat{D}_i = \{ \langle \dots, 0, \dots, \overset{i}{d},
\dots, 0, \dots \rangle \mid d \in D_{i^{-}} \} \cup \{ 1 \}.
\]
Then  $\widehat{D}_i$  is a complete sublattice of
 $\Pi^* ( D_i \mid i \in I )$ , and  $\widehat{D}_i$  is
isomorphic to  $D_i$ . Let  $\Theta_i$  be the restriction of
 $\Theta$  to  $\widehat{D}_i$ .

Since  $D_i \setminus \{0\}$  is complete-simple, so is  $\widehat{D}_i$ , and
hence  $\Theta_i$  is  $\omega$  or  $\iota$ . If
 $\Theta_i = \rho$  for all  $i \in I$ , then
 $\Theta = \omega$ . If there is an  $i \in I$ , such that
 $\Theta_i = \iota$ , then  $0 \equiv 1 \pmod{\Theta}$ , hence
 $\Theta = \iota$ .
\end{proof}

The Main Theorem follows easily from Theorems \ref{T:P*}
and~\ref{T:P*a}.

\begin{thebibliography}{9}
\bibitem{sF90}
Soo-Key Foo, Lattice Constructions, Ph.D. thesis, University
of Winnebago, Winnebago, MN, December, 1990.
\bibitem{gM68}

```


George~A. Menuhin, Universal algebra. D.~Van Nostrand,
Princeton, 1968.

\bibitem{eM57}
Ernest~T. Moynahan, On a problem of M. Stone, Acta Math.
Acad. Sci. Hungar. \textbf{8} (1957), 455--460.

\bibitem{eM57a}
\bysame, Ideals and congruence relations in lattices. II,
Magyar Tud. Akad. Mat. Fiz. Oszt. K\{"o}zl.
\textbf{9} (1957), 417--434 (Hungarian).

\bibitem{fR82}
Ferenc~R. Richardson, General lattice theory. Mir, Moscow,
expanded and revised ed., 1982 (Russian).

\end{thebibliography}
\end{document}

A CONSTRUCTION OF COMPLETE-SIMPLE DISTRIBUTIVE LATTICES

GEORGE A. MENUHIN

ABSTRACT. In this note we prove that there exist *complete-simple distributive lattices*, that is, complete distributive lattices in which there are only two complete congruences.

1. INTRODUCTION

In this note we prove the following result:

Main Theorem. *There exists an infinite complete distributive lattice K with only the two trivial complete congruence relations.*

2. THE $D^{(2)}$ CONSTRUCTION

For the basic notation in lattice theory and universal algebra, see Ferenc R. Richardson [5] and George A. Menuhin [2]. We start with some definitions:

Definition 1. Let V be a complete lattice, and let $\mathfrak{p} = [u, v]$ be an interval of V . Then \mathfrak{p} is called *complete-prime* if the following three conditions are satisfied:

- (1) u is meet-irreducible but u is *not* completely meet-irreducible;
- (2) v is join-irreducible but v is *not* completely join-irreducible;
- (3) $[u, v]$ is a complete-simple lattice.

Now we prove the following result:

Lemma 1. *Let D be a complete distributive lattice satisfying conditions (1) and (2). Then $D^{(2)}$ is a sublattice of D^2 ; hence $D^{(2)}$ is a lattice, and $D^{(2)}$ is a complete distributive lattice satisfying conditions (1) and (2).*

Proof. By conditions (1) and (2), $D^{(2)}$ is a sublattice of D^2 . Hence, $D^{(2)}$ is a lattice.

Since $D^{(2)}$ is a sublattice of a distributive lattice, $D^{(2)}$ is a distributive lattice. Using the characterization of standard ideals in Ernest T. Moynahan [3], $D^{(2)}$ has a zero and a unit element, namely, $\langle 0, 0 \rangle$ and $\langle 1, 1 \rangle$. To show that $D^{(2)}$ is complete, let $\emptyset \neq A \subseteq D^{(2)}$, and let $a = \bigvee A$ in D^2 . If $a \in D^{(2)}$, then $a = \bigvee A$ in $D^{(2)}$; otherwise, a is of the form $\langle b, 1 \rangle$ for some $b \in D$ with $b < 1$. Now $\bigvee A = \langle 1, 1 \rangle$ in D^2 and the dual argument shows that $\bigwedge A$ also exists in D^2 . Hence D is complete. Conditions (1) and (2) are obvious for $D^{(2)}$. \square

Corollary 1. *If D is complete-prime, then so is $D^{(2)}$.*

Date: March 15, 2024.

2020 Mathematics Subject Classification. Primary: 06B10; Secondary: 06D05.

Key words and phrases. Complete lattice, distributive lattice, complete congruence, congruence lattice.

Research supported by the NSF under grant number 23466.

The motivation for the following result comes from Soo-Key Foo [1].

Lemma 2. *Let Θ be a complete congruence relation of $D^{(2)}$ such that*

$$(2.1) \quad \langle 1, d \rangle \equiv \langle 1, 1 \rangle \pmod{\Theta},$$

for some $d \in D$ with $d < 1$. Then $\Theta = \iota$.

Proof. Let Θ be a complete congruence relation of $D^{(2)}$ satisfying (2.1). Then $\Theta = \iota$. \square

3. THE Π^* CONSTRUCTION

The following construction is crucial to our proof of the Main Theorem:

Definition 2. Let D_i , for $i \in I$, be complete distributive lattices satisfying condition (2). Their Π^* product is defined as follows:

$$\Pi^*(D_i \mid i \in I) = \Pi(D_i^- \mid i \in I) + 1;$$

that is, $\Pi^*(D_i \mid i \in I)$ is $\Pi(D_i^- \mid i \in I)$ with a new unit element.

Notation. If $i \in I$ and $d \in D_i^-$, then

$$\langle \dots, 0, \dots, \overset{i}{d}, \dots, 0, \dots \rangle$$

is the element of $\Pi^*(D_i \mid i \in I)$ whose i -th component is d and all the other components are 0.

See also Ernest T. Moynahan [4]. Next we verify:

Theorem 1. *Let D_i , for $i \in I$, be complete distributive lattices satisfying condition (2). Let Θ be a complete congruence relation on $\Pi^*(D_i \mid i \in I)$. If there exist $i \in I$ and $d \in D_i$ with $d < 1_i$ such that for all $d \leq c < 1_i$,*

$$(3.1) \quad \langle \dots, 0, \dots, \overset{i}{d}, \dots, 0, \dots \rangle \equiv \langle \dots, 0, \dots, \overset{i}{c}, \dots, 0, \dots \rangle \pmod{\Theta},$$

then $\Theta = \iota$.

Proof. Since

$$(3.2) \quad \langle \dots, 0, \dots, \overset{i}{d}, \dots, 0, \dots \rangle \equiv \langle \dots, 0, \dots, \overset{i}{c}, \dots, 0, \dots \rangle \pmod{\Theta},$$

and Θ is a complete congruence relation, it follows from condition (3) that

$$(3.3) \quad \begin{aligned} & \langle \dots, \overset{i}{d}, \dots, 0, \dots \rangle \\ & \equiv \bigvee (\langle \dots, 0, \dots, \overset{i}{c}, \dots, 0, \dots \rangle \mid d \leq c < 1) \equiv 1 \pmod{\Theta}. \end{aligned}$$

Let $j \in I$ for $j \neq i$, and let $a \in D_j^-$. Meeting both sides of the congruence (3.2) with $\langle \dots, 0, \dots, \overset{j}{a}, \dots, 0, \dots \rangle$, we obtain

$$(3.4) \quad \begin{aligned} 0 &= \langle \dots, 0, \dots, \overset{i}{d}, \dots, 0, \dots \rangle \wedge \langle \dots, 0, \dots, \overset{j}{a}, \dots, 0, \dots \rangle \\ &\equiv \langle \dots, 0, \dots, \overset{j}{a}, \dots, 0, \dots \rangle \pmod{\Theta}. \end{aligned}$$

Using the completeness of Θ and (3.4), we get:

$$0 \equiv \bigvee (\langle \dots, 0, \dots, \overset{j}{a}, \dots, 0, \dots \rangle \mid a \in D_j^-) = 1 \pmod{\Theta},$$

hence $\Theta = \iota$. \square

Theorem 2. *Let D_i for $i \in I$ be complete distributive lattices satisfying conditions (2) and (3). Then $\Pi^*(D_i \mid i \in I)$ also satisfies conditions (2) and (3).*

Proof. Let Θ be a complete congruence on $\Pi^*(D_i \mid i \in I)$. Let $i \in I$. Define

$$\widehat{D}_i = \{\langle \dots, 0, \dots, \overset{i}{d}, \dots, 0, \dots \rangle \mid d \in D_i^- \} \cup \{1\}.$$

Then \widehat{D}_i is a complete sublattice of $\Pi^*(D_i \mid i \in I)$, and \widehat{D}_i is isomorphic to D_i . Let Θ_i be the restriction of Θ to \widehat{D}_i .

Since D_i is complete-simple, so is \widehat{D}_i , and hence Θ_i is ω or ι . If $\Theta_i = \rho$ for all $i \in I$, then $\Theta = \omega$. If there is an $i \in I$, such that $\Theta_i = \iota$, then $0 \equiv 1 \pmod{\Theta}$, hence $\Theta = \iota$. \square

The Main Theorem follows easily from Theorems 1 and 2.

REFERENCES

- [1] Soo-Key Foo, Lattice Constructions, Ph.D. thesis, University of Winnebago, Winnebago, MN, December, 1990.
- [2] George A. Menuhin, Universal algebra. D. Van Nostrand, Princeton, 1968.
- [3] Ernest T. Moynahan, On a problem of M. Stone, Acta Math. Acad. Sci. Hungar. **8** (1957), 455–460.
- [4] ———, Ideals and congruence relations in lattices. II, Magyar Tud. Akad. Mat. Fiz. Oszt. Közl. **9** (1957), 417–434 (Hungarian).
- [5] Ferenc R. Richardson, General lattice theory. Mir, Moscow, expanded and revised ed., 1982 (Russian).

COMPUTER SCIENCE DEPARTMENT, UNIVERSITY OF WINNEBAGO, WINNEBAGO, MN 53714

Email address: menuhin@ccw.uwinnebago.edu

URL: <http://math.uwinnebago.edu/homepages/menuhin/>

11.4 Article templates

In this section, we create a template to be used for `amsart` articles. Open the template with a text editor and save it under a different name. You can then start to write your new article using the new file, without having to remember the details governing the preamble and the top matter.

Create the template, which contains a customized preamble and top matter with sample bibliographic items, in several steps.

Step 1 In your text editor, open the `amsart.tpl` document from the `samples` folder (see page 5) and save it in your work subfolder as `myams.tpl`. Alternatively, type in the lines as shown in this section.

The first few lines of the file are

```
% Sample file: amsart.tpl

%Preamble
\documentclass{amsart}
\usepackage{amssymb,latexsym}
```

Notice the use of commented out lines (lines that start with `%`) that have been added as comments about the file.

Edit line 1 to read

```
% Template file: myams.tpl
```

The lines

```
\documentclass{amsart}
\usepackage{amssymb,latexsym}
```

specify the `amsart` document class and the use of the `amssymb` and `latexsym` packages to gain access, by name, to all the symbols listed in Appendices A and B.

Step 2 After the `\usepackage` command, there are sets of proclamation definitions corresponding to the examples in Section 4.4.2. Choose Option 5 for `myams.tpl` by deleting all the lines related to the other options. You are left with the lines

```
%Theorems, corollaries, lemmas, and propositions, in the
%most emphatic (plain) style. All are numbered separately.
%There is a Main Theorem in the most emphatic (plain)
%style, unnumbered. There are definitions, in the less
%emphatic(definition) style. There are notations, in the
%least emphatic (remark) style, unnumbered.
```

```

\theoremstyle{plain}
\newtheorem{theorem}{Theorem}
\newtheorem{corollary}{Corollary}
\newtheorem*{main}{Main Theorem}
\newtheorem{lemma}{Lemma}
\newtheorem{proposition}{Proposition}

```

```

\theoremstyle{definition}
\newtheorem{definition}{Definition}

```

```

\theoremstyle{remark}
\newtheorem*{notation}{Notation}

```

Step 3 Two more choices are presented. You can have either one or two authors—for more complex situations; see Section 11.2.4. For the `myams.tpl` template, choose one author by deleting everything between

```
%Two authors
```

```
and
```

```
%End Two authors
```

You are left with

```

\begin{document}
%One author
\title[shorttitle]{titleline1\\
                  titleline2}
\author{name}
\address{line1\\
         line2\\
         line3}
\email{name@address}
\urladdr{http://homepage}
\thanks{thanks}
%End one author

\keywords{keywords}
\subjclass[2020]{Primary: subject; Secondary: subject}
\date{date}

\begin{abstract}
  abstract

```

```

\end{abstract}
\maketitle

\begin{thebibliography}{99}

\end{thebibliography}
\end{document}

```

In the top matter, fill in your own personal information. For instance, I edited

```

\author{name}

to read

\author{George~Gr~"atzer}

```

I also edited `\address`, `\email`, `\urladdr`, and `\thanks`. After the editing, I had the following:

```

%top matter
\title[shorttitle]{titleline1\\
                    titleline2}
\author{George~Gr~"atzer}
\address{University of Manitoba\\
          Department of Mathematics\\
          Winnipeg, MB R3T 2N2\\
          Canada}
\email{gratzer@ms.umanitoba.ca}
\urladdr{http://server.maths.umanitoba.ca/homepages/gratzer/}
\thanks{Research supported by the NSERC of Canada.}

\keywords{keywords}
\subjclass[2010]{Primary: subject; Secondary: subject}
\date{date}

\begin{abstract}
  abstract
\end{abstract}
\maketitle

\begin{thebibliography}{99}

\end{thebibliography}
\end{document}

```

Since this template is meant to be used for all my future articles, I do not edit the lines that change from article to article (`\title`, `\keywords`, and so on).

Remember that the short title is for running heads, the title shown at the top of every odd-numbered page other than the title page. If the title of your article is only one line long, delete the separation mark `\\` and the second line, except for the closing brace. If the full title of your article is short, delete `[shorttitle]`.

Now save `myams.tpl`. I saved my template under the name `ggamsart.tpl` (in the `samples` folder; see page 5). You can also make an additional template with two authors to be used as a template for joint articles. Note that at the end of the template, just before the line `\end{document}`, there are two lines:

```
\begin{thebibliography}{99}

\end{thebibliography}
```

The argument of `\begin{thebibliography}` should be 9 if there are fewer than 10 references, 99 with 10–99 references, and so forth. We discuss how to format bibliographic items in 10.5.1. The templates for bibliographic items are listed after the `\end{document}` line.

To make sure that you do not overwrite your template, I recommend that you make it read-only. How you do this depends on your computer’s operating system.

You should modify the template you create in this section to the template of the journal you submit your article to. In the `samples` folder, you find the AMS template for the Proceedings of the AMS, called `amsproc.tpl`.

11.5 Options

The `amsart` document class supports a number of options, affecting many attributes. For each attribute there is a *default value* that is used if a value is not specified.

Font size

```
Options:  9pt
          10pt  default
          11pt
          12pt
```

This option declares the default font size. You may want to use the 12pt option for proofreading:

```
\documentclass[12pt]{amsart}
```

Remember, however, that changing the font size changes the line breaks, so changing the 12pt option back to 10pt may require that you make some adjustments in the text (see Section 1.4).

Paper size

Options: letterpaper (8.5 inches by 11 inches) *default*
 legalpaper (8.5 inches by 14 inches)
 a4paper (210 mm by 297 mm)

Equations and equation numbers

A number of options deal with the placement of equations and equation numbers.

Options: leqno *default*
 reqno

By default, equation numbers are placed on the left, the default `leqno` option. The `reqno` option places the equation numbers on the right.

Option: fleqn

This option positions equations a fixed distance from the left margin rather than centering them. The `fleqn` option is typically used in conjunction with the `reqno` option. Here is how an equation looks with the `fleqn` and `reqno` options:

$$\int_0^{\pi} \sin x \, dx = 2 \tag{1}$$

typed as

```
\begin{equation}\label{E:firstInt}
  \int_{0}^{\pi} \sin x \, dx = 2
\end{equation}
```

Options: tbtags
 centertags *default*

The `tbtags` option uses *top-or-bottom tags* for a `split` environment, that is, it places the equation number level with the last line if numbers are on the right, or level with the first line if the numbers are on the left:

$$\begin{aligned} (1) \quad f &= (x_1 x_2 x_3 x_4 x_5 x_6)^2 \\ &= (x_1 x_2 x_3 x_4 x_5 + x_1 x_3 x_4 x_5 x_6 + x_1 x_2 x_4 x_5 x_6 + x_1 x_2 x_3 x_5 x_6)^2 \\ &= (x_1 x_2 x_3 x_4 + x_1 x_2 x_3 x_5 + x_1 x_2 x_4 x_5 + x_1 x_3 x_4 x_5)^2 \end{aligned}$$

The `centertags` option (the default) vertically centers the equation number in a `split` subsidiary math environment.

Limits

Options: intlimits
 nointlimits *default*

The `intlimits` option places the subscripts and superscripts of integral symbols above and below the integral symbol rather than on the side in a displayed math formula—with this option you can use the `\nolimit` command to disable the option for one integral. The `nointlimits` option positions the subscripts and superscripts of integral symbols on the side.

Options: `sumlimits` *default*
 `nosumlimits`

The `sumlimits` option places the subscripts and superscripts of large operators, such as \sum , \prod , \coprod , \otimes , \oplus , above and below the large operator in a displayed math formula. `nosumlimits` positions them on the side (see Table 7.7 and Appendix A.6.1).

Options: `namelimits` *default*
 `nonamelimits`

The `namelimits` option places the subscripts and superscripts of operators with limits such as `det`, `inf`, `lim`, `max`, `min`, and so on, above and below the operator in a displayed math formula. `nonamelimits` positions them on the side (see Tables 7.4, 7.5, and Appendix A.6).

Two-sided printing

Options: `twoside` *default*
 `oneside`

The `twoside` option formats the output for printing on both sides of the paper. The alternative is the `oneside` option. This option influences running heads, the placement of page numbers, and so on.

Two-column printing

Options: `twocolumn`
 `onecolumn` *default*

The `twocolumn` option typesets the document in two columns.

Title page

Options: `titlepage`
 `notitlepage` *default*

The `titlepage` option creates a separate title page including the abstract. The `notitlepage` option splits the top matter between the first and last pages of the typeset article.

Draft

Options: `draft`
 `final` *default*

The `draft` option prints a slug in the margin next to each line that is too wide. The `final` option does not. Note that this option is passed on to some packages, such as `graphicx`.

Fonts

Option: `noamsfonts`

With this option, the document class does not load the packages necessary for the use of the AMSFonts font set.

No math

Option: `nomath`

By default, `amsart` loads the `amsmath` package (which, in turn, loads three more math packages). If you want to use the title page and related features without the math features, you can use the `nomath` option. Probably obsolete.

11.6 The AMS packages

If you follow the recommendation of this book and begin each article with

```
\documentclass{amsart}
\usepackage{amssymb,latexsym}
```

then you can safely ignore most of the information in this section. There are two minor exceptions, the packages `amsxtra` and `upref`.

However, if you use a document class that does not load the same packages that `amsart` loads, then you have to load the packages needed for your work. Typically, you have

```
\usepackage{amsmath,amsfonts,amsthm}
\usepackage{amssymb,latexsym}
```

as a minimum. Make sure that `amsmath` loads before `amsthm`.

The AMS distribution contains many packages that can be loaded together or by themselves.

Math enhancements

amsmath The primary math enhancement package, which loads the four packages, `amsgen`, `amsbsy`, `amsopn`, and `amstext`.

- amsbsy** Provides two commands for the use of bold math symbols, `\boldsymbol` and `\pmb` (see Section 8.4.3).
- amscd** Commands for creating simple commutative diagrams (see Section 9.8). Probably obsolete.
- amsgen** An auxiliary package that is never invoked directly. It is loaded by all the AMS math packages (except for `upref`).
- amsopn** Provides operator names and also the `\DeclareMathOperator` command for defining new ones (see Section 7.6).
- amstext** Defines the `\text` command and redefines commands such as `\textrm` and `\textbf` to behave like the `\text` command (see Section 7.4.6).
- amsxtra** Provides the “sp” math accents (see Sections 7.7 and A.7) and loads the `amsmath` package.
- upref** Ensures that the `\ref` command always produces upright numbers.

AMSFonts

- amsfonts** Contains the basic commands needed to utilize the AMSFonts. It also defines the `\mathfrak` command which makes the Euler Fraktur math alphabet available (see Section 8.4.2). If you want to use the 12pt document class option, then you must also load the `exscale` package:

```
\usepackage{exscale}
```

- amssymb** Defines the symbol names for `amsfonts`. It loads `amsfonts`.
- eucal** Replaces the calligraphic math alphabet with the Euler Script math alphabet (see Section 8.4.2). If you load it with the option `mathscr`, as in

```
\usepackage[mathscr]{eucal}
```

then both the `\mathscr` and the `\mathcal` commands are available, so you can have both \mathcal{C} and \mathscr{C} , typed as

```
 $\mathcal{C}$  and  $\mathscr{C}$ 
```

- eufrak** Defines the Euler Fraktur math alphabet (see Section 8.4.2).

Loading packages

`amsart` contains code to provide more flexible formatting of proclamations and the proof environment (see Sections 4.4.2 and 4.5). By loading the `amsthm` package you can add this functionality to a non-AMS document class. The `amsthm` package loads the `amsgen` package.

`amsart` loads four packages from the math enhancements group, the `amsmath`, `amsbsy`, `amstext`, `amsopn`, and `amsgen` packages, and the `amsfonts` package from the AMSFonts group.

A typical article using the legacy `article` document class and the AMS enhancements would normally have

```
\documentclass{article}
\usepackage{amsmath}% math enhancements
\usepackage{amssymb,latexsym}% AMSFonts and LaTeX symbol names
\usepackage{amsthm}% proclamations with style
```

and perhaps the following:

```
\usepackage{eucal}% Euler Script
```

Note that it is not critical for you to remember which packages load others. No harm is done if you type

```
\usepackage{amsmath}
\usepackage{amsbsy}
```

The `amsbsy` package is loaded by the `amsmath` package, and the

```
\usepackage{amsbsy}
```

line is ignored by \LaTeX .

All the math related options of `amsart` (see Section 11.5) are also options of the `amsmath` package. So, for instance, if you want the equation numbers on the right, load `amsmath` with the `reqno` option:

```
\usepackage[reqno]{amsmath}
```

Multiple indices

The AMS distribution also contains the package `amsidx` for creating multiple indices for `amsbook`. This package is discussed in Section 18.5.

11.7 Legacy documents

About 40 years ago, L^AT_EX introduced the very important concept of document classes (separation of content from styling) and presented five document classes: `article`, `report`, `book`, `letter`, `slides`. Today, `letter` and `slides` are obsolete. The `book` document class is mostly used as a foundation for custom document classes. The document class for this book, for example, is about 50% of the `book` code and 50% custom code. We discuss the `book` document class in detail in Chapter 19.

The `article` and `report` document classes are very similar, we'll briefly discuss `article`.

There are only four commands for the top matter: `\title`, `\author`, `\thanks`, and `\date`. We are familiar with them, except that the `\author` command now has to handle a lot of information: the authors name, affiliation, research support, email, and so on. To avoid these complications, use the `amsart` document class.

PART VI

PDF Documents



Hyperlinks

12.1 Hyperlinks for L^AT_EX

12.1.1 Hyperlinks

With Adobe Acrobat Pro (and many other PDF editors), you can place *hyperlinks* in PDF documents. Clicking on a hyperlink, you jump to another location in the same document, to an electronic document, or to a Web site. For instance, in the table of contents, you can put a hyperlink to Chapter 3, so that clicking on it takes you to Chapter 3. Adobe Acrobat's help system has ample information on how to set up links. Of course, it is tedious to set hyperlinks one at a time in your PDF file. Would it not be nice if hyperlinks corresponding to cross-references were set automatically? For instance, clicking on Lemma 6 in

This follows from Lemma 6 and the relevant definitions.

would cause the display to jump to the page containing Lemma 6.

Sebastian Rahtz's `hyperref` package does just that.

12.1.2 Using hyperref

You invoke the `hyperref` package with the command

```
\usepackage{hyperref}
```

as the *last* `\usepackage` line in the preamble of your \LaTeX document. (There are a very few exceptions.) Figure 12.1 shows a page fragment from a mathematical article with hyperlinks to some sections, theorems, and citations automatically created by `hyperref`.

The construction of the uniquely complemented lattice representing a given monoid is introduced in [Section 4](#). It is based on V. Koubek and J. Sichler [12]. [Section 5](#) proves that this construct has many simple sublattices. Finally, in [Section 6](#), we put all these pieces together to construct the lattice L for the Main Theorem.

To prove [Theorem 1](#), we need a different construction, which is presented in [Section 7](#).

Figure 12.1: The `hyperref` package with the `\autoref` command.

See [Section 12.1.5](#) for the `autoref` command. To see how `hyperref` works, look up the `art2-ref.tex` article in the `samples` folder. It is the sample article `art2.tex` enhanced with the `hyperref` package. The article uses the `hyperref` options

```
pagebackref,colorlinks,bookmarks=true
```

See [Section 12.1.3](#) for the `pagebackref` and `colorlinks` options. The third option, `bookmarks=true`, is discussed in [Section 12.1.4](#).

Copy `art2-ref.tex` into the work folder and typeset it twice. The PDF file created for you has some of the `hyperref` features, but not all. Open the pdf file with Adobe Reader. Look at the left pane. Bookmarks is a table of contents of the article, with links to the named sections. Pages is a thumbnail sketches of the pages, with links to them.



Tip Use the `hyperref` package with caution, it may cause unexpected problems. For instance, it inserts a blank page 510 in this book. Also, the `hyperref` package with the `backref` option aborts the typesetting if there is a `\verb` command in the references.

12.1.3 backref and colorlinks

A useful addition to `hyperref` is David Carlisle's `backref` package. It is invoked as an option of `hyperref`:

```
\usepackage[backref]{hyperref}
```

- [12] V. Koubek and J. Sichler, *Universality of small lattice varieties*, Proc. Amer. Math. Soc. **91** (1984), 19–24. [1](#), [2](#), [4](#), [4.2](#), [4.3](#), [4.3](#), [4.3](#), [7](#)
- [13] H. Lakser, *Simple sublattices of free products of lattices*, Abstract, Notices Amer. Math. Soc. **19** (1972), A 509. [1](#), [3](#), [3](#)

Figure 12.2: The hyperref package with the backref option.

The items in your bibliography will be followed by a list of sections in which the bibliographic reference is cited. Each number printed after the cited reference becomes a hyperlink to the relevant section. Alternatively, you can use the `pagebackref` option, which produces a list of page numbers. Figure 12.2 shows a page fragment from a bibliography displaying lists of section numbers. `backref` can be used to check if all items in the bibliography have actually been referenced in the article. Any reference that has not been cited does not have a page listed.

Another popular option is `colorlinks`, which colors the text of the links instead of underlining them.



Tip Make sure that in the bibliography any two items are separated by a blank line and the last item is separated from `\end{bibliography}` by a blank line. Otherwise, you may get very confusing error messages with `\backref`.

If your bibliography is created by BIB_TE_X, see Chapter 16, it has these blank lines. Don't edit them out!

12.1.4 Bookmarks

An important navigational feature of Adobe Reader is the ability to set and use bookmarks. If you choose View>Navigation Tabs>Bookmarks in Adobe Reader, the navigation pane opens up showing the bookmarks.

The `hyperref` package option `bookmarks=true` makes bookmarks from the sectioning commands of the L^AT_EX document, thereby producing a table of contents even if the document had none. You can invoke all these options together:

```
\usepackage[backref,colorlinks,bookmarks=true]{hyperref}
```

Typesetting your L^AT_EX document with the `bookmarks=true` option produces an out file, which contains entries such as

```
\BOOKMARK [1] [-]{section.1}{\376\377\0001\000.\000\040
\000I\000n\000t\000r\000o\000d\000u\000c\000t\000i\000o\000n}{}% 1
```

Once you have produced the final version of your document, you should edit this file to make sure that it contains no L^AT_EX code. Math formulas in titles create havoc. Once this file has been edited, add the line

```
\let\WriteBookmarks\relax
```

at the start of the file to prevent it from being overwritten.

12.1.5 Additional commands

The `hyperref` package has dozens of commands and parameters, but we will discuss only four more commands.

Preventing links

If you do not want a `\ref` or `\pageref` command to appear as a link, you can use their *-ed forms, `\ref*` and `\pageref*`.

Long links

An often heard complaint is that in the link Theorem 6, only the 6 can be clicked to activate the link, and it is too short. `hyperref` provides the `\autoref` command to help out. Instead of

```
Theorem~\ref{T:new}
```

you can simply type

```
\autoref{T:new}
```

and `hyperref` will provide the word Theorem so that the link becomes [Theorem 6](#). The names supported by the `\autoref` command are listed in Table 12.1.

For my own use, I redefine:

```
\renewcommand{\chaptername}{Chapter}
\renewcommand{\sectionname}{Section}
\renewcommand{\subsectionname}{Section}
\renewcommand{\subsubsectionname}{Section}
```

External links

External links can be links to websites or other files that are on the Internet. Use the

```
\href{address}{text}
```

command to typeset *text* and make it into a link to the Web address (URL).

Command	Meaning
<code>\figurename</code>	Figure
<code>\tablename</code>	Table
<code>\partname</code>	Part
<code>\appendixname</code>	Appendix
<code>\equationname</code>	Equation
<code>\Itemname</code>	item
<code>\chaptername</code>	chapter
<code>\sectionname</code>	section
<code>\subsectionname</code>	subsection
<code>\subsubsectionname</code>	subsubsection
<code>\paragraphname</code>	paragraph
<code>\Hfootnotename</code>	footnote
<code>\AMSname</code>	Equation
<code>\theoremname</code>	Theorem

Table 12.1: Redefinable names supported by `\autoref`.

For instance, in your references, you may have

Writing mathematics with MathML.

`\href{https://developer.mozilla.org/en-US/docs/Learn/MathML}`

Then the last line of the address becomes a link and clicking on it takes you to the Web site. As an even fancier example, note the top matter command `\urladdr` (see Section 11.2.2) in `art2-ref.tex`:

`\urladdr{\href{http://math.uwinnebago.edu/menuhin/}`
`http://math.uwinnebago.edu/homepages/menuhin/}`

Then, as part of Menuhin’s address, you will find

<http://math.uwinnebago.edu/menuhin/>

Now clicking on the Web address will link to his Web page.

`hyperref`, of course, offers a lot more than I have presented here. For more detail, see the user manual and Section 2.4.6 of LC3.

Many of the problems mentioned can be avoided by using the `cleveref` package.

12.2 Line numbers

Sometimes, it is useful to show the line numbers in the PDF file you create. For instance, when I distribute the PDF file to the volunteers who read it, I provide line numbers—to facilitate reporting comments. It is easy to do this. In the preamble, add

```
\usepackage{lineno}
```

and add

```
\linenumbers
```

after the `\begin{document}`, or wherever you want the line numbering to begin. You may prefer

```
\modulolinenumbers[5]
```

which adds a line number to every fifth line; this is just as useful and less clustered.



Presentations

In Section 1.11, we describe how a *presentation* is a PDF file that you open with Adobe Reader. You can put it in full screen mode¹ (View>Full screen), and then project the presentation one page at a time by pressing the space bar or the arrow keys.

Remember overhead transparencies? If we want to see half of what is on the transparency, we cover up the bottom part so that only the top part is projected. This way we have control over what the audience sees and when. We sometimes used overlays: placing another transparency on top of the projected one to modify it by adding text or graphics.

In this chapter, we discuss Till Tantau's *beamer* package in more detail to help you prepare presentations with overlays and with stunning visual effects. *beamer* relies on other packages such as the *hyperref* package (see Section 12.1) to establish links, Till Tantau's *PGF* package for creating graphics, Uwe Kern's *xcolor* package for coloring, the *AMS* packages for formatting math formulas and defining declarations, and some others.

¹In Acrobat Reader, open File>Document Properties and check mark Full Screen Mode. Then the PDF document automatically opens in full-screen mode.

The documentation for these packages runs to about a thousand pages. The good news is that you can use beamer “out of the box”. You only have to learn about **20 commands**—this is more than the **four commands** we had to learn in Section 1.11 but still an easily manageable task. So we set ourselves in this chapter a modest goal, using beamer “out of the box”. It is amazing how much you can achieve with a small investment of your time.

A warning: be very careful to follow the rules for beamer. If the code is perfect, beamer works great. If it is not, you are on your own, there are no beamer error messages.

13.1 *Quick and dirty* beamer

We convert the article `art1.tex` (in the `samples` folder) to a beamer presentation. We will remove some commands that are appropriate for an article but not for a presentation and add some commands that are specific to presentations. This will not produce a very good presentation. Nevertheless, the conversion is a really quick introduction to some basic beamer concepts.

1. First changes

Open `art1.tex`, save a copy in the work folder. Make the following changes in the preamble and top matter:

1.1 Change the first line to

```
%Introductory beamer presentation: quickbeamer1.tex
```

1.2 Change the documentclass to beamer.

1.3 Delete the `\usepackage` and `\newtheorem` lines—beamer loads the necessary packages and defines this declaration.

Also delete `\subclass`, `\keywords`, `\email`, and `\urladdr` lines.

1.4 Change the `\address` to `\institute`—this is the beamer command for address.

1.5 Delete the abstract environment—this is not needed for the presentation.

Here is the new version of the preamble and top matter.

```
%Introductory beamer presentation: quickbeamer1.tex
\documentclass{beamer}
\begin{document}
\title{A technical result for congruences of finite lattices}
\author{G. Gr\"atzer}
\institute{Department of Mathematics\\
University of Manitoba\\
```

```

Winnipeg, MB R3T 2N2\\
Canada}
\date{March 21, 2024}
\maketitle

```

2. Changes in the body

- 2.1** If you follow these instructions with your own article, delete the proof environment, but not the contents, that is, delete the two lines

```

\begin{proof}
\end{proof}

```

The proof environment (see Section 3.3.2) is not suitable for presentations because an environment can only be used within a frame, and proofs are typically longer. (The article `art1.tex` has no proof environment.)

- 2.2** Cut the presentation into *frames* (pages, transparencies, foils) with the `frame` environments. After each `\begin{frame}`, we put a `\frametitle` command. The argument of the command is the “title” for the frame, displayed prominently at the top of the display.

It would be tedious to give you precise instructions on how to do this; instead, refer to the `quickbeamer1.tex` document (in the `samples` folder) for all the `frame` environments and `\frametitle` commands we added.

Typeset the result. This is the purest translation of `art1.tex` to beamer. That’s it, enjoy your first presentation.

3. Improving the presentation

Now you make some small changes to `quickbeamer1.tex` to utilize beamer’s powerful effects. Changes 2 and 3 are quite dramatic.

Save `quickbeamer1.tex` with the name `docquickbeamer2.tex` in the work folder.

- 3.1** To add a section, place the `section` command before the frame. We add three sections: `Introduction`, `The result`, and `The proof`. We correspondingly remove the `\frametitle` commands. We add a frame after the title frame:

```

\begin{frame}
  \frametitle{Outline}
  \tableofcontents[pausesections]
\end{frame}

```

to create the table of contents; we call it `Outline`. Note how the `Outline` is created, displayed in three slides.

3.2 Add `\usetheme{Berkeley}` after the `documentclass` line. Typeset and be stunned.

3.3 Change `\maketitle` to

```
\begin{frame}
  \titlepage
\end{frame}
```

Make sure that the last (sub)section is followed by a frame, otherwise it will be missing from the table of contents.

3.4 Add this frame after the titlepage frame:

```
\begin{frame}
\frametitle{Outline}
\tableofcontents[pausesections]
\end{frame}
```

This creates a table of contents frame, with the section titles appearing one at a time.

3.5 Replace all instances of `{equation}` by `{equation*}` or `[and]`. In a presentation a reference to another frame is not recommended so equations should not be numbered. You might as well delete all the `\label` commands since these are not needed either.

3.5 Change the bibliographic reference to

See also Ernest T. Moynahan, 1957.

Turning `quickbeamer2.tex` into a PDF file will get you a much prettier presentation. Four pages of the new presentation are displayed in Figures 13.1 and 13.2—unfortunately, without the nice colors.

4. Adjusting the navigation

Looking at Figures 13.1 and 13.2, we see that the Berkeley theme turns the sidebar into a navigation device and the section titles produce the table of contents (the Outline frames). Remember to typeset a few times!

Article `art2.tex` has three sections. They are navigation tools! The sidebar lists all the sections and highlights the section we are in. By clicking on the title of a section, the presentation jumps there.

But two problems come to light.

4.1 The title of the presentation is too long for the sidebar.

4.2 There is no need to repeat the author's name in the sidebar.

A technical result for congruences of finite lattices

G. Grätzer

Department of Mathematics
University of Manitoba
Winnipeg, MB R3T 2N2
Canada

March 21, 2024



Introduction

In some recent research, G. Czédli and I, spent quite an effort in proving that some equivalence relations on a planar semimodular lattice are congruences. The number of cases we had to consider was dramatically cut by the following result.



Figure 13.1: quickbeamer1 presentation, slides 1 and 2

The result

Theorem

Let L be a finite lattice. Let δ be an equivalence relation on L with intervals as equivalence classes. Then δ is a congruence relation iff the following condition and its dual hold:

If x is covered by $y, z \in L$ and $x \equiv y \pmod{\delta}$, then $z \equiv y \vee z \pmod{\delta}$.

Proof

We prove the join-substitution property:

If $x \leq y$ and $x \equiv y \pmod{\delta}$, then

$$x \vee z \equiv y \vee z \pmod{\delta}.$$

Let $U = [x, y \vee z]$. We induct on length U , the length of U .

Figure 13.2: quickbeamer1 presentation, slides 3 and 4.

To correct these, save the file `quickbeamer1.tex` as `quickbeamer2.tex` in the work folder. The edited version is in the `samples` folder along with the PDF file. `endenumerate`

We make two changes.

4.a Change the `\title` command to

```
\title[Congruences of finite lattices]
{A technical result\
for congruences of finite lattices}
```

and the second `\section` command to

```
\section[Construction]{The  $\Pi^*$  construction}
```

The bracketed parts are the short versions used in the sidebar.

4.b Change the `\author` command to

```
\author[] {G. Gr\"atzer}
```

The short version of the `author` command is blank, so the author's name will not be displayed in the sidebar.

Figure 13.3 shows slide 7 of the `quickbeamer2` presentation—this corresponds to page 4 of the `quickbeamer1` presentation; the Outline accounts for the difference in the slide number. Note how all the deficiencies listed above have been corrected. Compare slide 7 of this presentation with the Berkeley theme in Figure 13.3 and with the Warsaw theme in Figure 13.4.² Themes are discussed in Section 13.5.

13.2 Baby beamers

In this section, we discuss the most important beamer concepts: overlays (and its many variants), blocks, links, columns, and coloring. Do not be overwhelmed by the many new commands. The pause command may suffice for most presentations.

13.2.1 Overlays

The Outline frame of the `quickbeamer2` presentation created three slides in the PDF file. Observe how each slide, from the second on, completely overlaps the previous one, making it appear that the previous one “stayed put” and an additional line is displayed “on top of it”. In beamer terminology these slides are *overlays*. The subsequent sections discuss many variants.

beamer has many commands for creating overlays. We start with some examples of the commands `\pause`, `\only`, and `\onslide`.

²See them in color in the sample file `6colorpages`.

Congruences of finite lattices

Introduction

The result

The proof

Theorem

Let L be a finite lattice.

Let δ be an equivalence relation on L with intervals as equivalence classes.

Then δ is a congruence relation iff the following condition and its dual hold: If x is covered by $y, z \in L$ and $x \equiv y \pmod{\delta}$, then $z \equiv y \vee z \pmod{\delta}$.

Navigation icons: back, forward, search, etc.

Figure 13.3: quickbeamer2 presentation, slide 6 with Berkeley theme.

Introduction

The result

The proof

Theorem

Let L be a finite lattice.

Let δ be an equivalence relation on L with intervals as equivalence classes.

Then δ is a congruence relation iff the following condition and its dual hold: If x is covered by $y, z \in L$ and $x \equiv y \pmod{\delta}$, then $z \equiv y \vee z \pmod{\delta}$.

Navigation icons: back, forward, search, etc.

Congruences of finite lattices

Figure 13.4: quickbeamer2 presentation, slide 6 with Warsaw theme.

We introduce overlays with some presentations. The first, `babybeamer1`, introduces the `\pause` command for creating overlays.

```
%Introductory beamer presentation: babybeamer1.tex
\documentclass{beamer}
\usetheme{Berkeley}

\begin{document}
\title{Congruences of finite lattices}

\begin{frame}
  \titlepage
\end{frame}

\begin{frame}
\frametitle{Outline}
\tableofcontents[pausesections]
\end{frame}

\begin{frame}
In some recent research, G. Cz\'edli and I,
spent quite an effort in~proving
that some equivalence relations
on a planar\semimodular lattice are congruences.
\end{frame}

\begin{frame}
There was always the problem of having to distinguish
so many different cases when proving that

If  $x$  is covered by  $y, z$  in  $L$  and  $x \equiv y \pmod{\delta}$ ,
then  $z \equiv y \vee z \pmod{\delta}$ .

The number of cases we had to consider
was dramatically cut by the following result.
\end{frame}

\end{document}
```

produces the presentation of Figure 13.5.³ You can see how bad it is. First, it displays the first three lines, then these line jump up quite a bit, and underneath it displays the rest of the lines. A really jarring effect.

³See this in color in the sample file `6colorpages`.

To correct this, we add a pause command:

```
on a planar\\semimodular lattice are congruences.
```

```
\pause
```

```
There was always the problem of having to distinguish
```

Now the display is perfect; see Figure 13.6.⁴ In slide 2, the first three lines are displayed, and in slide 3, the rest of the lines “slide” in perfectly.



Rule: The `\pause` command

1. A frame may have many `\pause` commands.
2. The `\pause` command cannot be given within an AMS multiline math environment.

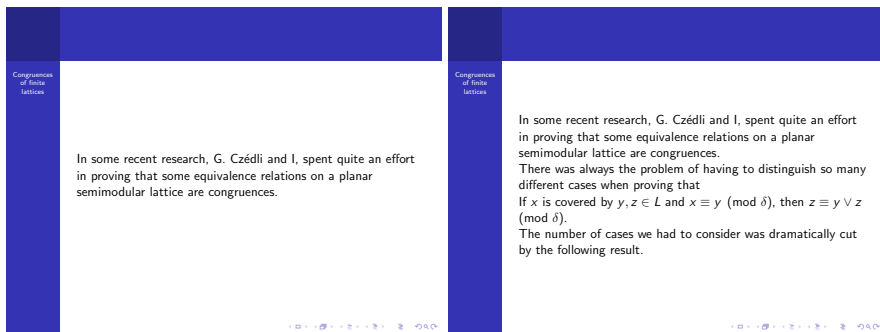


Figure 13.5: babybeamer1 presentation, slides 2 and 3

You move past a `\pause` command the same way as you get to the next frame, by pressing the space bar or the forward arrow key.

Using `\pause` commands, you can create many overlays, each containing a little more material on the slides. If this is all you need, skip to Section 13.2.7.

We could have coded the same presentation with the `\only` command:

```
\begin{frame}
\only<1,2>{In some recent research, G. Cz\'edli and I,
spent quite an effort in~proving
that some equivalence relations
on a planar\\semimodular lattice are congruences.}
\only<2>{There was always the problem of having to distinguish
```

⁴See it in color in the sample file 6colorpages.

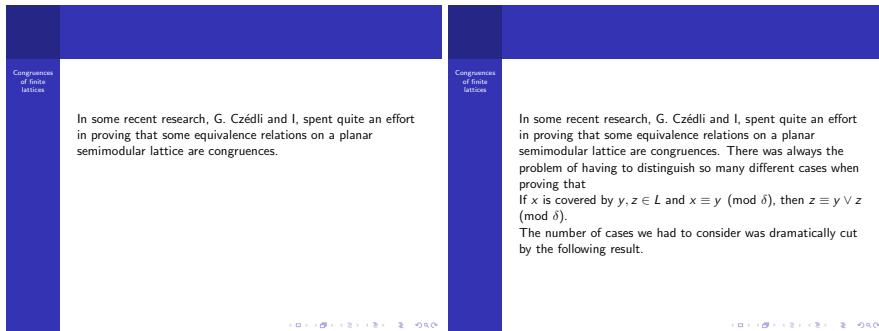


Figure 13.6: babybeamer1 presentation with pause, slides 2 and 3

so many different cases when proving that

If x is covered by $y, z \in L$ and $x \equiv y \pmod{\delta}$, then $z \equiv y \vee z \pmod{\delta}$.

The number of cases we had to consider was dramatically cut by the following result.

This produces two slightly different slides. The first ignores the second `\only` command and displays the line as appropriate to display one line. Slide 2 displays the two lines as appropriate to display two lines. As a result, the first line moves slightly up when passing from overlay 1 to overlay 2. The argument of an `\only` command is typeset only on the overlays specified. On the other overlays, it is ignored.

If instead of the `\only` command you use the `\onslide` command (on slide, get it?), as in

```
\begin{frame}
\onslide<1,2>{In some recent research, G. Cz\'edli and I,
spent quite an effort in~proving
that some equivalence relations
on a planar\semimodular lattice are congruences.}
\onslide<2>{There was always the problem of having to distinguish
so many different cases when proving that
```

If x is covered by $y, z \in L$ and $x \equiv y \pmod{\delta}$, then $z \equiv y \vee z \pmod{\delta}$.

```
The number of cases we had to consider
was dramatically cut by the following result.}
\end{frame}
```


then the first line of slide 2 completely overlaps the first line of slide 1, so it seems to stay put. The argument of the `\onslide` command is typeset on the slides specified and on the other slides it is typeset but invisible. This is the behavior you would want most often, but you may find that sometimes you prefer `\only`.

13.2.2 Understanding overlays

We introduced overlays in the previous section—probably the most important new concept for presentations. \LaTeX typesets the content of a frame, and the typeset material:

- appears on all overlays for the parts of the source (maybe all) not modified by any command with an overlay specification;
- appears only on the overlays specified and is ignored on the other overlays for the arguments of the `\only` commands;
- appears on the overlays specified and is typeset but made invisible on the other overlays for the arguments of the `\onslide` commands.

More on overlay specifications at the end of this section. Here are some illustrations.

Example 1

This is a very `\only<1>\{very, very\}` important concept.
`\only<1,2>\{To start the definition \dots\}`

will typeset slide 1 as

```
┌
│   This is a very very, very important concept. To start the definition ...
└
```

and will typeset slide 2 as

```
┌
│   This is a very important concept. To start the definition ...
└
```

Example 2

What is $2+2$? It is `\onslide<2>\{4\}`.
`\only<1>\{Can you figure it out?\}`

`\onslide<2>\{I hope you all got it right.\}`

will typeset overlay 1 as

```
┌
│   What is 2 + 2? It is . Can you figure it out?
└
```

and will typeset overlay 2 as

```
┌
    What is 2 + 2? It is 4.
└    I hope you all got it right.
```

Note that there is room in overlay 1 for the number 4.

Example 3

What is $2+2$?

```
\onslide<2>{It is  $4$ .}
```

Can you figure it out?

will typeset slide 1 as

```
┌
    What is 2 + 2?
└    Can you figure it out?
```

and will typeset slide 2 as

```
┌
    What is 2 + 2?
    It is 4.
└    Can you figure it out?
```

Note that there is room in slide 1 for the “missing” second line.

Slide specifications

The angle brackets contain the *slide specification*. Here are some more examples:

```
<1-2,4->    means all slides from 1 to 2, and all slides from 4 onwards.
<-3>        means all slides up to 3.
<2,4,6>     means slides 2, 4, and 6.
```

We have two slide specifications in the presentation `babybeamer2`: `<1,2>` and `<2>`. Maybe, `<1->` and `<2->` would be better, so that if you add a third slide you do not have to change these.

The command `\pause` can only take the simplest slide specification, a number. `\pause<3>` takes effect from overlay 3 on.

13.2.3 More on the `\only` and `\onslide` commands

The `\only` and `\onslide` commands can accomplish everything the `\pause` command can and a lot more.

The basic syntax

The syntax of `\only` is

```
\only<slide spec>{source}
```

where *slide spec* is the overlay specification and *source* is the code typeset by L^AT_EX.

A (partial) syntax of `\onslide` is

```
\onslide<slide spec>{source}
```

With the same syntax you can give overlay specifications to many commands, including `\textbf`, `\textit`, `\alert`—beamer’s alternative to the `\emph` command—and then the command is in effect only on the overlays specified. Here is a simple example.

```
\begin{frame}
\frametitle{Some background}
\textbf<1>{We start our discussion with some concepts.}
\textbf<2>{The first concept we introduce originates
         with Erd\H os.}
\end{frame}
\end{document}
```

So this frame (see Figure 13.7) produces two slides, each with two lines of text. On slide 1 the first line is bold, on slide 2 the second line is bold.

A different syntax

The command `\only` has an alternate syntax:

```
\only{source}<slide spec>
```

So

```
\only<1>{Can you figure it out?}
```

and

```
\only{Can you figure it out?}<1>
```

accomplish the same.

With this syntax, you can define your own commands that allow overlay specifications. For instance, using the command `\color{blue}` defined in Section 13.2.9, you can define the command

```
\newcommand{\myblue}{\only{\color{blue}}}
```

Then

```
\myblue<2>{Some more text}
```

will color the text blue on slide 2 only.

13.2.4 *Lists as overlays*

Lists may be presented one item at a time, as for example in the `babybeamer2` presentation in Figure 13.7 (in the `samples` folder) shows the first two overlays of a list. R. Padmanabhan appears on the first, R. Padmanabhan and Brian Davey appear on the second, and so on. This is accomplished simply by adding the overlay specification `<1->` to the item for R. Padmanabhan, the overlay specification `<2->` to the item for Brian Davey, and so on.

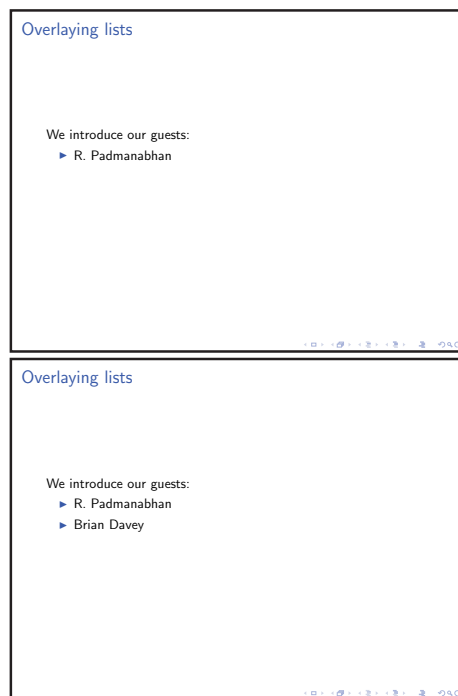


Figure 13.7: Slides 1 and 2 of the `babybeamer2` presentation.

```
%babybeamer2 presentation
\documentclass{beamer}
\begin{document}

\begin{frame}
\frametitle{Overlaying lists}

We introduce our guests:
\begin{itemize}
\item<1-> R. Padmanabhan
\item<2-> Brian Davey
\item<3-> Harry Lakser
\item<4-> Dick Koch
\end{itemize}
\end{frame}
\end{document}
```

Such an overlay structure is used so often that beamer has a shorthand for it, [$\langle + - \rangle$]. Here it is in babybeamer3.

```
%babybeamer3 presentation
\documentclass{beamer}
\begin{document}
\begin{frame}
\frametitle{Overlaying lists}
We introduce our guests:
\begin{itemize}[<+>-]
\item R. Padmanabhan
\item Brian Davey
\item Harry Lakser
\item Dick Koch
\end{itemize}
\end{frame}
\end{document}
```

This shorthand allows adding and reordering items without having to change slide specifications. Of course, if you do not want the items to appear in sequence, you have to use slide specifications.

13.2.5 Out-of-sequence slides

We now present an example of “out-of-sequence slides”. Look at Figure 13.8. I want to make this part of my presentation. First, I want to show the theorem, then illustrate

it with the diagram at the bottom. Finally, I present the proof in the middle. So I need three overlays.

The theorem is on all three overlays, 1, 2, 3. Its illustration is on overlays 2 and 3, leaving room for the proof that appears only on overlay 3. This is an example of “out-of-sequence overlays”. We code this in `babybeamer4` (in the `samples` folder).

Since declarations, proofs, and the `\includegraphics` command may all have overlay specifications, this seems easy to accomplish.

```
%babybeamer4 presentation, first try
\documentclass{beamer}
\begin{document}

\begin{frame}
\frametitle{Overlaying declarations and graphics}
\begin{theorem}<1->
Every finite distributive lattice can be embedded
in a boolean lattice.
\end{theorem}
\begin{proof}<3->
Use join-irreducible elements.
\end{proof}
\includegraphics<2->{cube}
\end{frame}
\end{document}
```

Theorem

Every finite distributive lattice can be embedded in a boolean lattice.

Proof.

Use join-irreducible elements.

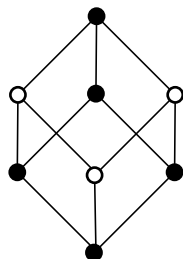


Figure 13.8: The slide to represent.

This does not work too well. On overlay 1 the theorem appears in the middle and then it jumps up to make room for the illustration. This is the same problem we encountered in the `babybeamer2` presentation in Section 13.2.1 and the solution is also the same, the use of the `\onslide` command. Replace the line

```
\includegraphics<2->{cube}
```

with

```
\onslide<2->{\includegraphics{cube}}
```

13.2.6 Blocks and slides

You can think of a theorem in `beamer` as the contents of the `theorem` environment with a heading and, optionally, with an slide specification, and with most themes—see Section 13.5—colorful visual highlighting; see Figures 13.3 and 13.4.

The `beamer` package provides the `block` environment that works the same way except that you name the block. The (partial) syntax of the `block` environment is

```
\begin{block}<slide spec>{title}  
source  
\end{block}
```

Blocks are shaped as theorems. If there is no title, you still need the braces. The slide specification is optional.

As an example, save `babybeamer4.tex` as `babybeamer4block.tex` in the work folder (also in the `samples` folder along with the PDF file) and replace the `theorem` environment with

```
\begin{block}<1->{Theorem}  
Every finite distributive lattice can be embedded  
in a boolean lattice.  
\end{block}
```

If you want a block of \LaTeX code with an overlay specification but with no title and no visual highlighting, then use one of the commands, `\onslide` and `\only`.

13.2.7 Links

A presentation is a PDF file, so it is not surprising that you can set links of various types in a `beamer` presentation. Just as the `hyperref` package helps us with hyperlinks in a PDF file (see Section 12.1), the `beamer` package allows us to conveniently set links in a presentation.

Some links are automatically provided. If you look closer at Figures 13.3 and 13.4, you see that the section titles are shown in the *sidebar*. In fact, the sidebar is a *navigation bar*. First, it shows which section you are in. Second, clicking on a section title takes you to that section.

Creating a link is a two-step process.

1. Name the place you want to link to.
2. Create a button with the property that clicking on it jumps you to the designated place.

To illustrate this process, we modify the presentation `babybeamer4`. Open the file `babybeamer5.tex` and save it as `babybeamer5.tex` in the work folder (the edited version is in the `samples` folder along with the PDF file).

1. Name the frame you want to link to by adding a label to the `\begin{frame}` line. In `babybeamer5`, add a label to the frame `fourguests`:

```
\begin{frame}[label=fourguests]
```

Labels of frames are also useful for selective typesetting of your presentation; see Section 13.6.

2. Add the following line to `babybeamer5`:

```
\hyperlink{fourguests<3>}%
{\beamergotobutton{Jump to third guest}}
```

This creates a link to the third overlay of the frame named `fourguests`, and creates a button, with the text: `Jump to third guest`. Clicking on this button will jump to the third slide of the frame `fourguests`.

3. To add variety to linking, include a new first frame:

```
\begin{frame}
\frametitle{First frame with a button}
Button example
```

```
Jumping to an overlay of a different frame
\bigskip
```

```
\hyperlink{fourguests<3>}%
{\beamergotobutton{Jump to third guest}}
\end{frame}
```

which has a button for jumping to the third overlay of the `fourguests` frame.

4. We also add a new third frame.

```
\begin{frame}
\frametitle{Third frame with a button}
Button example
```



```

Jumping to another frame
\bigskip

```

```

\hyperlink{fourguests}%
{\beamergotobutton{Jump to guest list}}
\end{frame}

```

with a button, with the text `Jump to guest list`. Clicking on this button will jump to the second frame, overlay not specified (defaults to 1).

5. Add a fourth frame,

```

\begin{frame}
\frametitle{Hidden link}
\hyperlink{fourguests}{Jumping to the guest list}
\end{frame}

```

introducing another version of the `\hyperlink` command:

```

\hyperlink{fourguests}{Jumping to the guest list}

```

which typesets the second argument as regular text, making it an *invisible link*. However, you may notice that the cursor changes when it hovers over the link. For instance, you may want to link the use of a concept to its earlier definition, where you also need a button for the return jump.

Here is `babybeamer5`:

```

%babybeamer5 presentation
\documentclass{beamer}
\begin{document}

\begin{frame}
\frametitle{First frame with a button}

```

Button example

```

Jumping to an overlay of a different frame
\bigskip

```

```

\hyperlink{fourguests<3>}%
{\beamergotobutton{Jump to third guest}}
\end{frame}

```

```

\begin{frame}[label=fourguests]

```

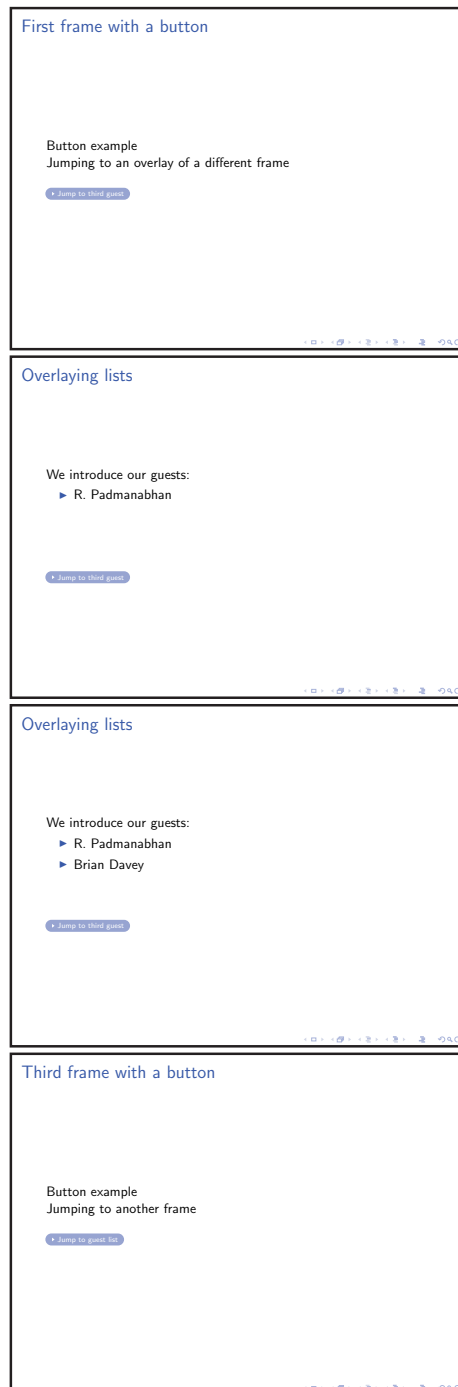


Figure 13.9: babybeamer5 presentation.

```

\frametitle{Overlaying lists}

We introduce our guests:
\begin{itemize}
\item<1-> R. Padmanabhan
\item<2-> Brian Davey
\item<3-> Harry Lakser
\item<4-> Dick Koch
\end{itemize}

\hyperlink{fourguests<3>}%
{\beamergotobutton{Jump to third guest}}
\end{frame}

\begin{frame}
\frametitle{Third frame with a button}
Button example

Jumping to another frame
\bigskip

\hyperlink{fourguests}%
{\beamergotobutton{Jump to guest list}}
\end{frame}

\begin{frame}
\frametitle{Hidden link}

\hyperlink{fourguests}{Jumping to the guest list}
\end{frame}
\end{document}

```

Figure 13.9 shows all these buttons. We do not show slides 3 and 4 of frame 2 and frame 4, where the button is invisible.

13.2.8 Columns

Often, it is useful to put the display into columns. A simple illustration is given in `babybeamer6`:

```

%babybeamer6 presentation
\documentclass{beamer}
\begin{document}

```

```

\begin{frame}
\frametitle{Columns, top alignment}

\begin{columns}[t]
\begin{column}{2in}
Is it true that there is no new result
on the Congruence Lattice Characterization Problem?
\end{column}
\begin{column}{2in}
F. Wehrung found a distributive algebraic lattice that
cannot be represented as the congruence lattice
of a lattice.
\end{column}
\end{columns}
\end{frame}
\end{document}

```

The environment is `columns`. It has an optional argument for alignment, `t` for top, `c` for center, and `b` for bottom. The columns, usually two, are both in the `column` environment; the width of the column is in the argument; it can be given as a measurement—`2in` in the example—or relative to the width of the whole frame as `0.4\textwidth`. Figure 13.10 shows the `babybeamer6` presentation.

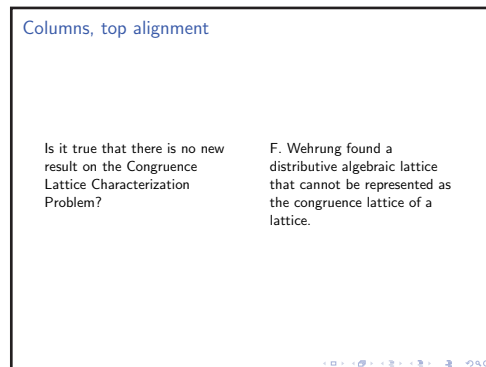


Figure 13.10: `babybeamer6` presentation.

13.2.9 Coloring

L^AT_EX’s job is to produce articles and books that contain text, math formulas, and graphics. Such publications—with the exception of textbooks—cannot afford color printing. Presentations are different. If you prepare a color presentation, it will project in color.

Nevertheless, the color commands are of limited use even for presentations. You

probably use the color scheme of the chosen theme (see Section 13.5), and have limited opportunity to color things yourself. If you do, be very careful, too much color distracts from the presentation but judicious use of color—say, for highlighting a word or phrase—may be very effective.

beamer uses the sophisticated xcolor package of Uwe Kern. It colors by specifying the color model: rgb (red, green, blue), or cmyk (cyan, magenta, yellow, black), or gray (black and white)—there are many more models to choose from—and how much of each color you want to mix.

So `\color[rgb]{0,1,0}` paints everything—within its scope—green. You can color some text green with the command

```
\textcolor[rgb]{0,1,0}{This text is green.}
```

There are seventeen predefined colors: red, green, blue, cyan, magenta, yellow, orange, violet, purple, brown, pink, olive, black, darkgray, gray, lightgray, and white. With the proper options, there are hundreds more. So the previous command could also be given as

```
\textcolor{green}{This text is green.}
```

or as

```
{\color{green}This text is green.}
```

To pretty things up, you can use `\colorbox{green}{Green box}`, which puts the argument in a green box and `\fcolorbox{red}{green}{Green box}`, which also adds a red frame.

xcolor is automatically loaded by beamer. To make sure that xcolor is loaded with the options desired, you have to include these options in the preamble in the

```
\documentclass{beamer}
```

line. For instance, to have the dvipsnam option for xcolor, invoke beamer with

```
\documentclass[xcolor=dvipsnam]{beamer}
```

You can also mix predefined colors:

```
{\color{green!40!yellow} This text is of what color?}
```

which sets the text 40% green and 60% yellow.

There are commands for defining colors and color sets, as well as for coloring the background, frames, and hyperlinks (see Uwe Kern, *Extending L^AT_EX's color facilities: the xcolor package* [28]).

Here is a simple illustration:

```

%babybeamer7 presentation
\documentclass{beamer}
\begin{document}
\setbeamercolor{normal text}{bg=yellow!15}
\begin{frame}
\frametitle{Colors}

\begin{columns}[t]
  \begin{column}{2in}
{\color{red}Is it true that there is no new result
on the Congruence Lattice Characterization Problem?}
  \end{column}
  \begin{column}{2in}
{\color{green}F. Wehrung found a distributive
algebraic lattice that cannot be represented
as the congruence lattice of a lattice.}
  \end{column}
\end{columns}
\end{frame}
\setbeamercolor{normal text}{bg=green!15}

\begin{frame}
\frametitle{Colors fading out}

We introduce our guests:
\begin{itemize}
\item {\color{red}R. Padmanabhan}
\item {\color{red!60!white}Brian Davey}
\item {\color{red!40!white}Harry Lakser}
\item {\color{red!20!white}Dick Koch}
\end{itemize}
\end{frame}
\end{document}

```

The command

```
\setbeamercolor{normal text}{bg=yellow!15}
```

sets the background color to light (15%) yellow. In the first column, the text is red, in the second, green. Set the foreground with `fg=`.

The background of the second frame is light green. The four participants are in lighter and lighter shades of red.

Figure 13.11 shows the frames of the `thebabybeamer7` presentation.⁵

⁵See them in color in the sample file `6colorpages`.



Figure 13.11: babybeamer7 presentation.

13.3 The structure of a presentation

The structure of your presentation is, by and large, determined by the sectioning commands: `\section` and `\subsection`. For a very long lecture there may also be `\part` commands. The argument of any of these commands may have a short version for the navigational side bar (see Section 13.1).

The sectioning commands used in a beamer presentation look the same as they do for articles and books, but they play a different role. They do not display a section title, but they add an entry to the table of contents. They also act as place markers in the sense that if you click on the title of a section in a navigation bar, then you will jump to the *frame following* the section command.



Rule: Sectioning commands

1. Sectioning commands can only be placed between frames.
2. There must be a frame following the last sectioning command.
3. For a long (sub)section title, use `\breakhere` to break a line.

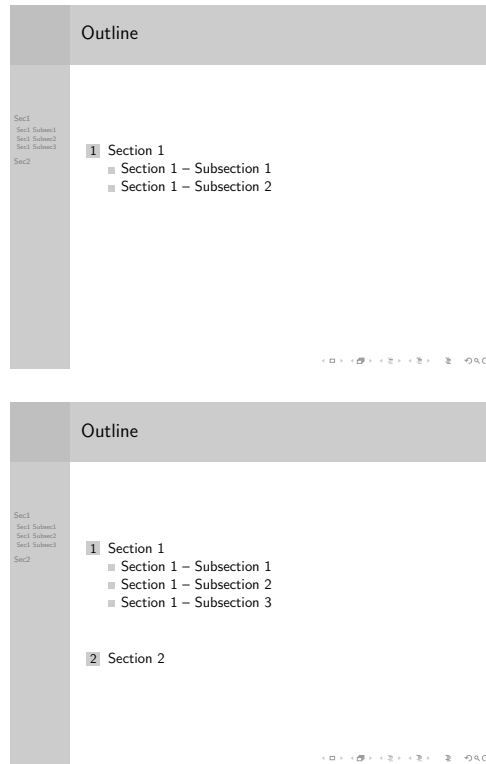


Figure 13.12: beamerstructure1 presentation, pages 3 and 5.

4. The optional short versions are for the navigation bar.

These are illustrated with `beamerstructure1`; see Figure 13.12. The line `\tableofcontents[pausessections, pausesubsections]`

causes the table of contents to appear a line at a time. This command may also be used without an option or only with one, `pausessections`.

The second page shown in Figure 13.12 is the table of contents. The page is about half filled with only five listed items, so no more than 10 sections and subsections would fit. There should be fewer.

```
% beamerstructure1 presentation
\documentclass{beamer}
\usetheme{Berkeley}

\begin{document}
```



```
\begin{frame}
\frametitle{Outline}

\tableofcontents[pausessections, pausesubsections]
\end{frame}

\section[Sec1]{Section 1}
\begin{frame}
\frametitle{Section 1}

Text of Section 1
\end{frame}

\subsection[Sec1 Subsec1]{Section 1 -- Subsection 1}
\begin{frame}
\frametitle{Section 1\\Subsection 1}

Text of Section 1, Subsection 1
\end{frame}

\subsection[Sec1 Subsec2]{Section 1 -- Subsection 2}
\begin{frame}
\frametitle{Section 1\\Subsection 2}

Text of Section 1, Subsection 2
\end{frame}

\subsection[Sec1 Subsec3]{Section 1 -- Subsection 3}
\begin{frame}
\frametitle{Section 1\\Subsection 3}

Text of Section 1, Subsection 3
\end{frame}

\section[Sec2]{Section 2}
\begin{frame}
\frametitle{Section 2}

Text of Section 2
\end{frame}
\end{document}
```

13.3.1 Longer presentations

Longer presentations may need parts and a more complicated table of contents. I will not discuss these topics, but the presentation `beamerstructure2` (in the `samples` folder) illustrates the use of parts and some other features. I added some comments to point these out. See Figure 13.13 for two sample pages of this presentation.

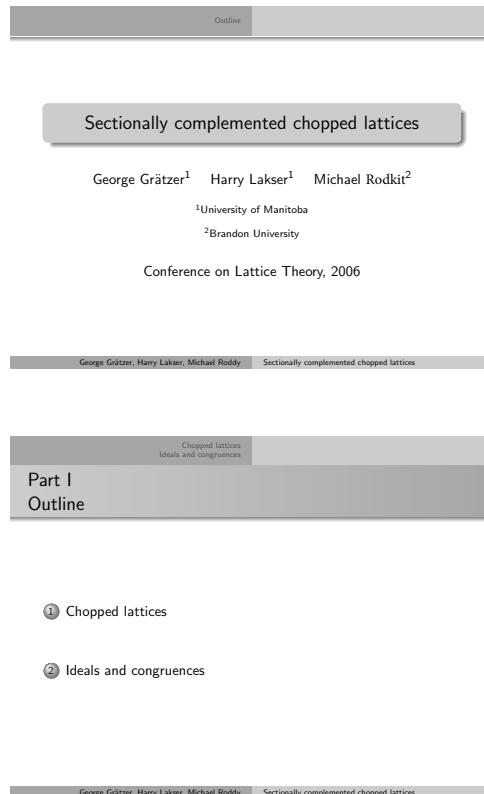


Figure 13.13: `beamerstructure2` presentation, slides 1 and 10.

13.3.2 Navigation symbols

The more structure you have in a presentation, the more you may appreciate the navigation icons shown by default on each page in the last line on the right.

The icons are:

- the slide,
 - the frame,
 - the section,
 - the presentation icons,
- each surrounded by a left and a right arrow,
- the appendix,
 - the back and forward icons (circular arrows),
 - the search icon (a magnifying glass).

If you decide not to have them, as in the presentation `beamerstructure2`, then give the following command in the preamble:

```
\setbeamer{navigation symbols}{}
```

13.4 Notes

You can place notes in your presentation to remind yourself of what you want to say in addition to what is being projected. A note is placed in the presentation as the argument of the `\note` command, as in

```
\note{This is really difficult to compute.}
```

By default, notes are not shown in the presentation. If you invoke `beamer` with

```
\documentclass[notes=show]{beamer}
```

then the notes pages are included. The command

```
\documentclass[notes=show, trans]{beamer}
```

produces transparencies with notes, and

```
\documentclass[notes=only]{beamer}
```

produces only the note pages, one note page for every overlay of a frame with a note. To avoid this, print the output of

```
\documentclass[trans, notes=only]{beamer}
```

In addition to these examples, all the notes placed in a single frame are collected together on one note page. And a note between frames becomes a page on its own.

`beamer` does an excellent job of producing notes, for example; see Figure 13.14. In the upper-left corner, it displays precisely where we are in the structure of the presentation. The upper-right corner shows a small picture of the page to which the notes are attached.

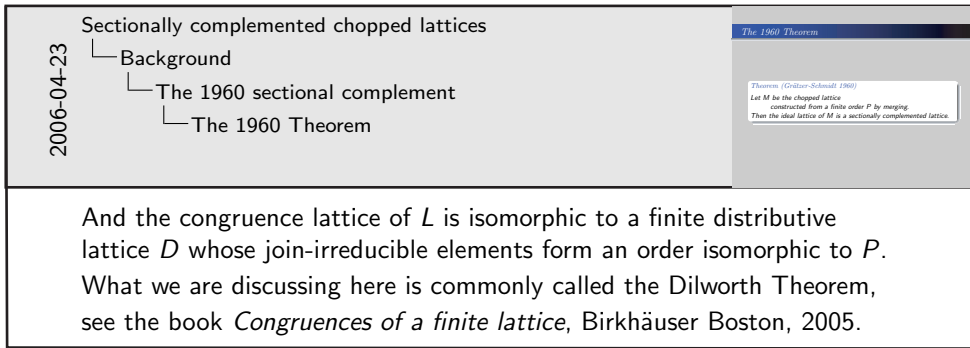


Figure 13.14: A note page.

13.5 Themes

If you look carefully at Figures 13.1, 13.2, and 13.13 (even better, if you look at the PDF files of these presentations), you see how every little detail of the presentation is provided by beamer. Figures 13.1 and 13.2 show a presentation style dominated by a dark blue headline and left sidebar, a complete navigation bar in the left sidebar, the name of the author(s) and the title repeated in every overlay, section numbers in colored squares, and so on.

The presentation in Figure 13.13 has no sidebars, section numbers are in colored circles, the navigation elements are in the headline. The display of lists and theorems (and other similar elements of a presentation) also vary a great deal.

You can achieve such detailed control over your presentation by defining all these elements yourself. beamer places dozens of commands at your disposal to make this possible. Or you can use a *presentation theme* that will do the job for you.

The command to name a presentation theme is `\usetheme{}`. The presentation beamerstructure2 uses the theme Warsaw (see Figure 13.13), so following the document class line type the command

```
\usetheme{Warsaw}
```

quickbeamer1 uses the theme Berkeley (see Figures 13.1 and 13.2) and so does beamerstructure1 (see Figure 13.12).

The presentation themes are in the `theme` subfolder of the `themes` folder of beamer. As of this writing, there are 26 of them, named after cities:

Presentation Themes:

Without Navigation Bars default, Bergen, Boadilla, Madrid, AnnArbor, CambridgeUS, Pittsburgh, Rochester.

With a Navigation Bar Antibes, JuanLesPins, Montpellier.

With a table of contents Sidebar Berkeley, PaloAlto, Goettingen, Marburg, Hannover.

With Mini Frame Navigation Berlin, Ilmenau, Dresden, Darmstadt, Frankfurt, Singapore, Szeged.

With Section and Subsection Table Copenhagen, Luebeck, Malmoe, Warsaw.

How do you choose a presentation theme? After the presentation is finished, try out the various themes. Ask yourself:

- Do sidebars take too much room away from my illustrations?
- Do stronger colors add to the presentation or do they distract from it?
- Do I want to use a navigation bar?

Answering these questions will narrow your choice.

The presentation theme defines all the colors, but you can alter them with the command `\usecolortheme{}`. You have a choice of albatross, beetle, crane, fly, and seagull.

For instance,

```
\usetheme{Warsaw}
\usecolortheme{seagull}
```

is a gray version of the Warsaw theme, appropriate for printing in black-and-white. In addition, you can further modify the “inner elements”, such as blocks, with

```
\usecolortheme{lily}
```

or orchid, or rose. You can modify the “outer elements”, such as headlines and sidebars, with

```
\usecolortheme{whale}
```

or seahorse, or dolphin. So you can have, for instance,

```
\usetheme{Warsaw}
\usecolortheme{lily}
\usecolortheme{whale}
```

This gives you 45 “out of the box” color schemes.

Similarly, font themes can also be specified, modifying the presentation theme, with the command `\usefonttheme{}`. You have the default and the following options:

```
professionalfonts
structurebold
structureitalicserif
structuresmallcapserif
```

13.6 Planning your presentation

Step 1 As a rule, your presentation is based on one or more of your articles. Collect them in one folder. Resolve naming conventions as necessary. There should be only one Fig1!

Step 2 Rewrite the article(s) to sketch out your presentation. The pages correspond to frames. A page should not have too many words, say, no more than 40. Replace your numbered theorems with named theorems. Never reference another page. Have few sections and subsections. Add a table of contents, which is a readable overview of the new article.

Step 3 Base the new presentation on a presentation in the `samples` folder, a sample presentation in beamer's `solution` folder, or on one of your own or of a colleague's older presentations. Turn the pages into frames.

Step 4 Design your frames and add frame titles. Completely disregard what we wrote in Section 2.3 (*the idea behind L^AT_EX is that you should concentrate on what you have to say and let L^AT_EX take care of the visual design*). The principle for presentations is: *You are completely responsible for the visual appearance of every frame and overlay.*

This is, of course, in addition to brevity and readability.



Tip Do not let L^AT_EX break your lines. Do it with the `\` command and keep words that belong together on the same line.

Step 5 Write notes to remind yourself what you want to say in your lecture that is not on the slides. Print the notes for your lecture.

Step 6 Build in flexibility. For instance, if you have four examples to illustrate a definition, put each one on a different frame or overlay, and add a link to each that skips the rest of the examples. Depending on your audience's understanding, show an example or two, and skip the rest. The same way, you may skip proof ideas and even topics.

Step 7 Prepare for the worst—the computer system may fail, but projectors seldom do—so print a set of transparencies for your lecture as a backup by invoking the option `trans` of the documentclass

```
\documentclass[trans]{beamer}
```

To print a *handout*, use the `handout` option

```
\documentclass[handout]{beamer}
```

Open the presentation in Acrobat Reader. In `Printer/Page Setup...` set landscape and 140% magnification. In the `Print` dialogue box in `Layout` choose two pages per sheet and print—assuming, of course, that you have a printer offering these options.

13.7 *What did I leave out?*

Since the beamer reference manual is 235 pages long, it is clear that this chapter covers maybe 10% of it.

For most presentations, you won't even need most of what I have included. If you read Sections 13.1, 13.2.1 and maybe Section 13.3, you should have enough for most presentations.

If you are in other fields, or if you are more ambitious, you may need more. For example, a computer scientist will want program listings in a verbatim environment. This is easy. Start your frame with

```
\begin{frame}{fragile}
```

and then you can use the verbatim environment.

If you want to include sounds or movies in your presentation, consult Till Tantau's *User's Guide to the Beamer Class* [43].

You can do very simple animation with what we have covered here. This is illustrated with the babybeamer8 presentation (in the samples folder).

```

babybeamer8 presentation
\documentclass{beamer}
\begin{document}

\begin{frame}

\includegraphics<1>{basem3-1}

\includegraphics<2>{basem3-2}

\includegraphics<3>{basem3-3}

\includegraphics<4>{basem3-4}
\end{frame}
\end{document}

```

The congruence generated by the dashed red line; see Figure 13.15, spreads in three steps, illustrating an interesting result. The animation is quite effective and instructive.

If you want to place such changing pictures lower in a frame, put them in the overprint environment.

I would recommend that you read Section 5 of Till Tantau, *User's Guide*, which has many good pointers about creating presentations.

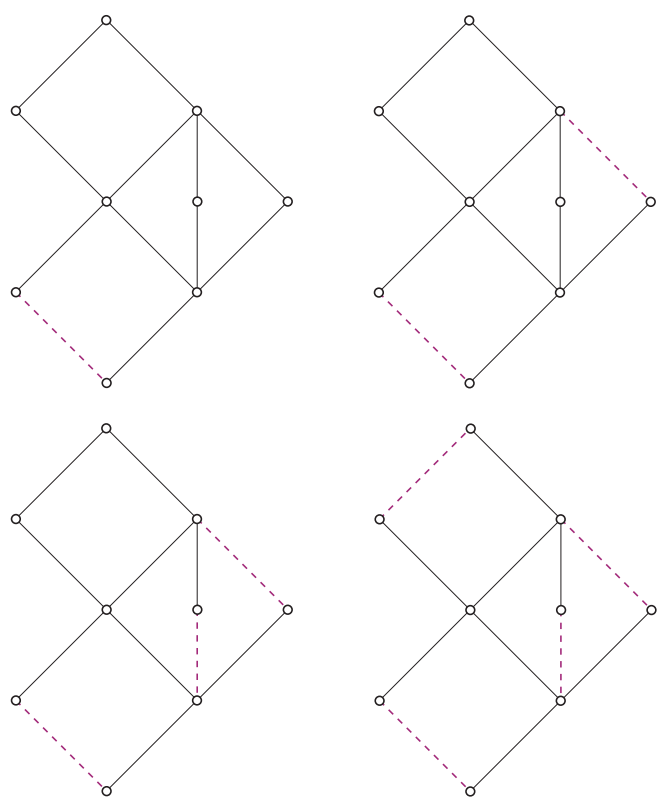


Figure 13.15: The four pictures of babybeamer8.



Illustrations

L^AT_EX can make beautiful text. It can also make beautiful illustrations. This chapter presents tools that enable a wide variety of such figures.

Creating illustrations is a complex topic with many subtleties. We use the onion approach: work on the easier outer concepts first, and then work towards the more complex inner ones later.

We use the TikZ package.¹ It is enabled by including

```
\usepackage{tikz}
```

in the preamble of the L^AT_EX document. Figures are then created using the `tikzpicture` environment, that is,

```
\begin{tikzpicture}
```

```
...
```

```
\end{tikzpicture}
```

There is a special document class that makes experimenting with TikZ particularly easy:

```
\documentclass[tikz]{standalone}
```

¹Michael Doob contributed the TikZ material in this chapter.

```

\begin{document}
  \begin{tikzpicture}
    ...
  \end{tikzpicture}
\end{document}

```

Insert the test code, apply L^AT_EX to the file, view the output, and play on!

14.1 *First steps: objects on the page*

We place objects on the page at user-given coordinates. They may be standard x - y coordinates (x, y) or polar coordinates $(\theta^\circ : r)$ (note the colon and the order in the latter case). It's legitimate to include units: $(1.5\text{in}, -24\text{pt})$. The default unit is cm . There are many commands to put objects on the page. The most frequent command is `\draw`; we can apply it to many different objects.

14.1.1 *Straight lines*

Some samples:

- A line joining two points:

```
\draw(2,3)--(3,2);
```

- A line joining two points by displacement:

```
\draw(2,3)-- +(1,-1);
```

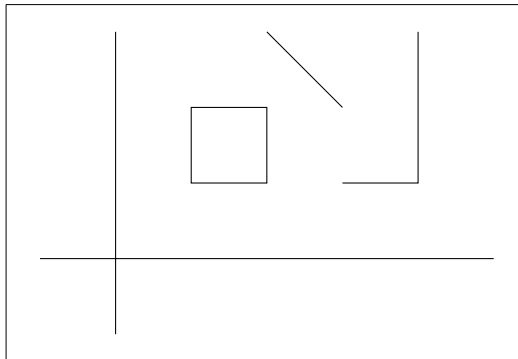
(note the plus sign). In this case $(1, -1)$ is the displacement from the first point, so this line drawn is identical to the previous one.

- A path:

```
\draw(2,0)--(3,0)--(3,1);
```

- A cycle:

```
\draw(0,0)--(0,1)--(1,1)--(0,1)--cycle;
```



A few lines

```

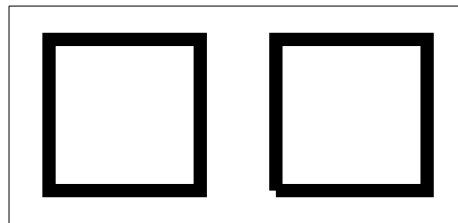
\draw (-1,0)--(5,0) (0,-1)--(0,3); % draw x-axis and y-axis
\draw (2,3)--(3,2);                % a diagonal line
\draw (3,1)--(4,1)--(4,3);          % a path with two edges
\draw (1,1)--(1,2)--(2,2)--(2,1)--cycle; % a unit square

```

**It figures ...**

- Comments are allowed (after all, this is L^AT_EX!).
- Every `\draw` is terminated by a semicolon; it's easy to forget!
- It is legal to concatenate more than one path within a single `\draw`.
`\draw(-3,0)--(3,0) (0,-1)--(0,1);`
draws a horizontal line, and a vertical line. It's a good way to see the x -axis and y -axis.

Look what happens when we use `cycle` to complete a cycle (on the left) and compare it to what happens when we repeat the first coordinate to complete the same cycle (on the right). Notice the notch in the lower left corner.



Completing the cycle

```

\draw [line width=5pt] (0,0)--(2,0)--(2,2)--(0,2)--cycle;
\draw [line width=5pt] (3,0)--(5,0)--(5,2)--(3,2)--(3,0);

```

**It figures ...**

- Insert the parameters changing the output of the `\draw` command between square brackets just after `\draw`. We changed the line width from the default value to 5 points (the default value is 0.4 points).
- Use `cycle` to complete a cycle rather than repeating the first coordinate.

The TikZ `\draw` command writes like a pen on paper. The command `\draw(0,1);` puts the pen down so that the current point is $(0, 1)$ (and does nothing). The command `\draw(0,1)--(1,1);` draws a line from the current point $(0, 1)$ to the point $(1, 1)$ (which then becomes the current point). The command

```
\draw(0,1)--(1,1)--(1,0);
```

uses the new current point and draws a line to (1, 0) and again updates the current point. The semicolon terminates the `\draw` command, that is, lifts up the pen.

It's sometimes useful to define new points relative to the current point.

```
\draw(1,1) -- +(1,0);
```

and

```
\draw(1,1) -- ++(1,0);
```

both draw a line from (1, 1) to (2, 1). In the former case, the current point is left at (1, 1), while in the latter case the current point is updated to (2, 1). In other words,

```
\draw(1,1) -- ++(1,0) -- ++(1,1);
```

is the same as

```
\draw(1,1) -- (2,1) -- (3,2);
```

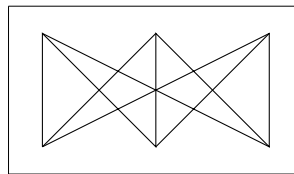
and

```
\draw(1,1) -- +(1,0) -- +(1,1);
```

is the same as

```
\draw(1,1) -- (2,1) (1,1) -- (2,2);
```

(with the current point remaining (1, 1)). The following example illustrates the two types of relative points:



Coordinate samples

```
\draw (0,0)-- ++(0,1)-- ++(1,-1)-- ++(0,1)
      -- ++(1,-1)-- ++(0,1)-- ++(-2,-1);
\draw (0,0)-- +(1,1) (1,0)-- +(1,1) (2,0)-- +(-2,1);
```

14.1.2 Parameters for lines

The characteristics of lines being drawn are changed by adding or adjusting the parameters. These most often appear within square brackets after the `\draw` command as a comma-separated list. For example,

```
\draw[style=dashed,color=gray,line width=2pt](0,1/8)--(2,1/8);
```

results in `-----`, a dashed, gray line.

TikZ allows a certain shorthand if there is no ambiguity. The same command could be given as

```
\draw[dashed,gray,line width=2pt](0,1/8)--(2,1/8);
```

since `dashed` can only be used to modify a line style, `gray` can only be used to modify a color, but a dimension may apply to many different parameters.








Name	Line width	
ultra thin	0.1pt	
very thin	0.2pt	
thin	0.4pt	
semithick	0.6pt	
thick	0.8pt	
very thick	1.2pt	
ultra thick	1.6pt	

Table 14.1: Predefined line widths























Line style	Line	Parameter	Arrow
solid		<code>[->]</code>	
dotted		<code>[<-]</code>	
loosely dotted		<code>[<->]</code>	
densely dotted		<code>[->>]</code>	
dashed		<code>[>>-]</code>	
loosely dashed		<code>[>>->>]</code>	
densely dashed		<code>[<<->>]</code>	
dashdotted		<code>[-]</code>	
loosely dashdotted		<code>[-]</code>	
densely dashdotted		<code>[-]</code>	
		<code>[<->]</code>	

Table 14.2: Line styles

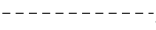
Table 14.3: Arrow parameters

Line widths

Line widths can be set specific values such as `line width=2pt`. There are some predefined values; see Table 14.1.

The command
`\draw[very thick] (0,0)--(1,1/4);`
results in the line: .

Line styles

Lines drawn using TikZ may be solid or have different combinations of dots and dashes. The command
`\draw[densely dashed] (0,1/8)--(2,1/8);`
results in the line: . See Table 14.2 for the predefined line styles.







Name	
default	
Latex	
Stealth	
Stealth [scale=1.5]	
Stealth [length=9pt,width=5pt]	

Table 14.4: Arrowheads

Arrows

A line is made into an arrow by a parameter that adds a head and/or a tail. The command

```
\draw[>->] (0,0.1)--(1,0.1);
```


gives . The parameters are similar to the shape of the arrow head and arrow tail; see Table 14.3.

Other arrow heads become available by loading an extra library:

```
\usetikzlibrary{arrows.meta}
```

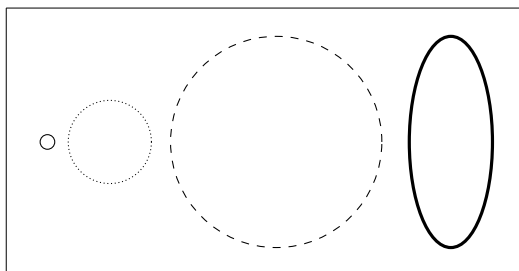
Then > can be replaced by Stealth or Latex in the parameter list:

```
\draw[Stealth-Stealth] (0,0.1)--(1,0.1);
```

results in . It is possible to roll your own by adjusting the length, width, or scale of an arrow head; see Table 14.4.

14.1.3 Circles, ellipses, dots, and parts thereof

To draw a circle or ellipse: follow `\draw` by the coordinates of the center, followed by `circle` or `ellipse`, followed by parameters setting the sizes of the radii. The line styles from Table 14.2 may be used.



Circles and an ellipse

```
\draw (0,0) circle [radius=5pt]; % small circle
\draw [densely dotted] (1.5,0) circle [radius=1]; % unit circle
\draw [dashed] (5.5,0) circle [radius=1in]; % large circle
```

**It figures ...**

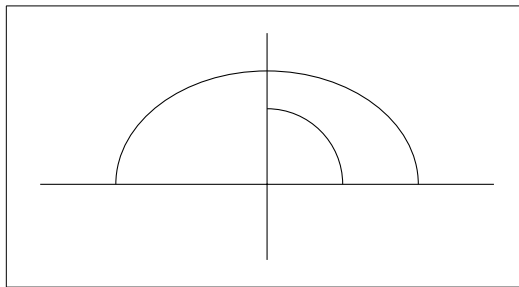
```
\draw [very thick] (9.7,0) ellipse [x radius=1cm, y radius=1in];
```

- The value assigned to a radius may be a real number followed by a unit of length such as cm, in, or pt. For the ellipse, the length of x radius is listed as cm even though it is the default. This is necessary: using units of length once in a command requires units of length be used everywhere in that command.
-

We form an arc from parts of circles or ellipses. We follow the `\draw` command by the point where the pen is put down. Then the parameters: starting angle (in degrees), the ending angle, and radius of the circle. For example, we draw the upper half of the unit circle:

```
\draw(1,0) arc [start angle=0, end angle=180, radius=1];.
```

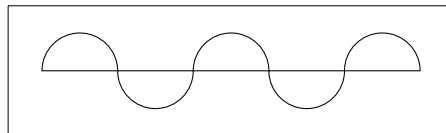
The syntax for the ellipse is almost identical: it assigns values to the x radius and the y radius.



Arcs from a circle and an ellipse

```
\draw (-3,0)--(3,0) (0,-1)--(0,2); % draw x-axis and y-axis
\draw (1,0) arc [start angle=0, end angle=90, radius=1];
\draw (2,0) arc [x radius=2, y radius=1.5,
start angle=0, end angle=180];
```

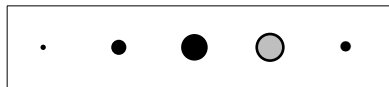
The syntax allows concatenation:



Concatenated arcs

```
\newcommand\Upper{arc [start angle=180, end angle=0, radius=1/2]}
\newcommand\Lower{arc [start angle=-180, end angle=0, radius=1/2]}
\draw (0,0) \Upper \Lower \Upper \Lower \Upper -- cycle;
```

A special type of circle is a dot, a small, filled circle that might represent a point on a plane or a vertex of a graph. Filling (closed) objects is common enough that it has its own command: `\fill`, which replaces `\draw`. Sometimes, it is desirable to fill a shape but retain its original outline. This could be done by using `\fill` on the object followed by `\draw` on the same object to restore the outline. Do both at once with `\filldraw`. Here are some dots of different sizes.



A few dots

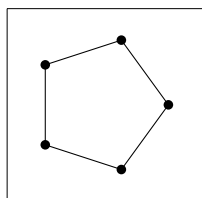
```
\fill (0,0) circle [radius=1pt]; % Tiny dot
\fill (1,0) circle [radius=.1]; % Normal dot
\fill (2,0) circle [radius=5pt]; % Larger dot
\filldraw [black, fill=lightgray, line width=1pt] (3,0)
  circle [radius=5pt]; % Larger dot filled
\node at (4,0) {$\bullet$}; % A math character
```



It figures ...

- The command `\node` adds text to a picture. Use `\node at (a,b) {...}` to put the text within the braces at the coordinate (a, b) . This command is described in detail in Section 14.3.
- It is good practice to have a consistent style from picture to picture inasmuch as it is possible. For example, at the beginning of the document define the command `\newcommand\mydot[1]{\fill #1 circle [radius=2pt]}`. Then `\mydot{(a,b)}` will put a dot with radius of two points at (a, b) in every `tikzpicture` environment.
- For a special circumstance where a different dot size is needed, `\renewcommand\mydot[1]{\fill #1 circle [radius=1.5pt]}` can be used to change the dot size within a particular `tikzpicture` environment. The change is local, that is, it disappears upon leaving the environment.

The next example puts several pieces together to draw a graph.

The graph C_5


```

\renewcommand\mydot[1]{\fill #1 circle [radius=2pt]}
\coordinate (P1) at (0:1);
\coordinate (P2) at (72:1);
\coordinate (P3) at (144:1);
\coordinate (P4) at (216:1);
\coordinate (P5) at (288:1);
\draw (P1)--(P2)--(P3)--(P4)--(P5)--cycle;
\mydot{(P1)}; \mydot{(P2)}; \mydot{(P3)};
\mydot{(P4)}; \mydot{(P5)};

```



It figures ...

- It's possible and useful to name each coordinate. In this example, each coordinate is used three times. Rewriting coordinates explicitly invites typographical errors. Also, we may reposition every reference to a coordinate with only one change.

14.1.4 The next step: curved lines between points

TikZ provides several constructions for creating curved lines between points. Some are short, straightforward, and cover many common cases, while others are more complicated but allow for a high degree of accuracy in more unusual cases. We describe both types of constructions.

The easy ones

Joining two given points by a curve is easy. Use

```
\draw(A) to (B);
```

to draw a straight line from (A) to (B), and


```
\draw[bend left] (A) to (B);
```

to draw a curve from (A) to (B) that bends to the left of the line joining those two points. The command

```
\draw[bend right] (A) to (B);
```


works similarly, by bending to the right. In each case, the angle at which the line leaves point (A) is symmetric with that at which the line enters point (B). The angle can be a parameter:

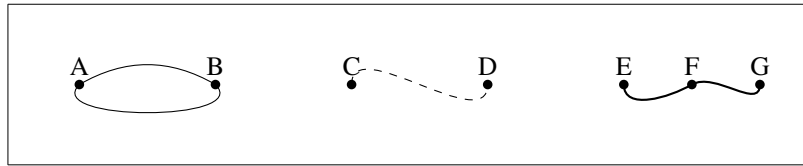
```
\draw[bend left=45] (A) to (B);
```

will create a curve that leaves (A) at an angle of 45° and enters (B) at an angle of 135° ; it looks like this: .

If we need greater control of those two angles, the angle the line leaves the starting point and the angle it enters the finishing point can be set separately. The command

```
\draw[out=270,in=90] (0,0.1) to (1,0.1);
```

results in .



Simple curves

```

\coordinate [label=A] (A) at (0,0);
\coordinate [label=B] (B) at (2,0);
\coordinate [label=C] (C) at (4,0);
\coordinate [label=D] (D) at (6,0);
\coordinate [label=E] (E) at (8,0);
\coordinate [label=F] (F) at (9,0);
\coordinate [label=G] (G) at (10,0);

\renewcommand\mydot[1]{\fill #1 circle [radius=2pt]}
\mydot{(A)}; \mydot{(B)}; \mydot{(C)}; \mydot{(D)};
\mydot{(E)}; \mydot{(F)}; \mydot{(G)};

\draw [bend left] (A) to (B);
\draw [bend right=135] (A) to (B);
\draw [dashed, out=90, in=270] (C) to (D);
\draw [thick, out=270, in=210] (E) to (F) to [out=30, in=270] (G);

```

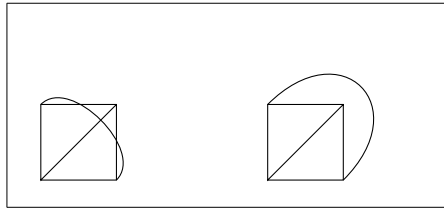
**It figures ...**

- We may define a coordinate and give it a nearby label in one command. This is helpful for keeping track of the points as the diagram is created. Label placement is described in detail in Section 14.3.
- Curves constructed this way may be concatenated. To make the curve smooth, the first in and the second out must differ by 180 (degrees).

The graph K_n has n vertices, each pair of which is joined by an edge. The task: draw K_4 so that edges do not cross each other. Put four vertices in a square and draw the edges of the square plus one diagonal. We could put in the last edge, say joining (B) and (D), using

```
\draw[bend left=90] (D) to (B);.
```

This is done for the image on the left in the next example and clearly doesn't work. To fix it, use the parameter `looseness`. The default value is `looseness=1`. As the value increases, the resulting curve is further away from the line joining (B) and (D). Increasing the value of `looseness` from 1 to 2.5 gives the desired result.

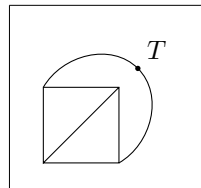


K_4 is planar

```
\coordinate (A) at (0,0); \coordinate (P) at (3,0);
\coordinate (B) at (1,0); \coordinate (Q) at (4,0);
\coordinate (C) at (1,1); \coordinate (R) at (4,1);
\coordinate (D) at (0,1); \coordinate (S) at (3,1);

\draw (A)--(B)--(C)--(D)--(A)--(C);
\draw [bend left=90] (D) to (B);
\draw (P)--(Q)--(R)--(S)--(P)--(R);
\draw [bend left=90, looseness=2.5] (S) to (Q);
```

There is another method to control the curve's shape: have it pass through a particular point. In the next example, the curve passes through an extra point T . The use of `in=135` for the first part of the curve and `out=315` for the second part (the two angles differ by 180°) ensures that the curve is smooth at T (of course the dot and label are ultimately unnecessary).



K_4 is planar

```
\coordinate (P) at (0,0);
\coordinate (Q) at (1,0);
\coordinate (R) at (1,1);
\coordinate (S) at (0,1);
\coordinate [label=above right:$T$] (T) at (1.25,1.25);
\fill (T) circle [radius=1pt];

\draw (P)--(Q)--(R)--(S)--(P)--(R);
\draw (S) to [out=60, in=135] (T) to [out=315, in=30] (Q);
```



It figures ...

- The `above right` in the `\coordinate` parameter determines the direction by which the label is offset. Also available: `above right`, `below left`, `below right`, `above`, `below`, `left`, and `right`.

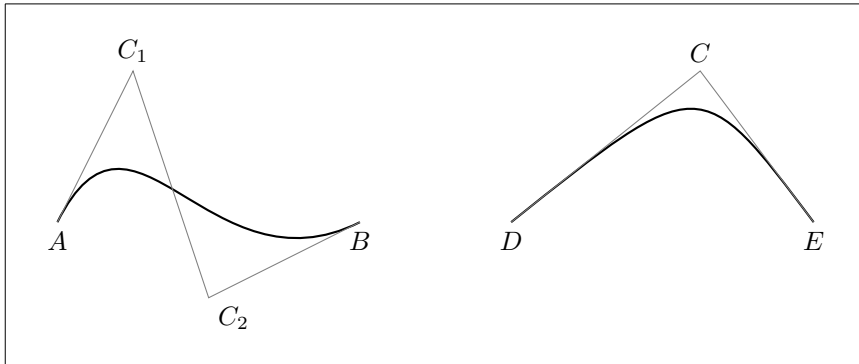
Bézier cubics

A Bézier cubic is a special and very flexible curve that is defined by four points: the starting point, two intermediate control points, and an ending point. A Bézier cubic with A as the starting point, C_1 and C_2 as control points, and B as the ending point is drawn using

```
\draw(A)..controls (C1) and (C2)..(B);.
```

There is an alternative form for the case where $C_1 = C_2 = C$ that draws a curve similar to (but not actually) a parabola:

```
\draw(D)..controls (C)..(E);.
```



Bézier cubics

```
\coordinate [label=below:$A$] (A) at (0,0);
\coordinate [label=below:$B$] (B) at (4,0);
\coordinate [label=above:$C_1$] (C1) at (1,2);
\coordinate [label=below right:$C_2$] (C2) at (2,-1);
\draw [thick] (A)..controls (C1) and (C2)..(B);
\draw [gray] (A)--(C1)--(C2)--(B);

\coordinate [label=below:$D$] (D) at (6,0);
\coordinate [label=below:$E$] (E) at (10,0);
\coordinate [label=above:$C$] (C) at (8.5,2);
\draw [thick] (D)..controls (C)..(E);
\draw [gray] (D)--(C)--(E);
```



It figures...

- The curve starts tangent to the line joining A and C_1 and ends tangent to the line joining B and C_2 .
- The curve is contained in the convex hull of $\{A, C_1, C_2, B\}$, that is, the smallest quadrilateral containing all four points.

For those with a mathematical bent: the Bézier curve starting at A , ending at B , with control points C_1 and C_2 is defined by

$$B(t) = (1-t)^3A + 3t(1-t)^2C_1 + 3t^2(1-t)C_2 + t^3B \quad (0 \leq t \leq 1).$$

We then have

$$B'(t) = 3(1-t)^2(C_1 - A) + 6t(1-t)(C_2 - C_1) + 3t^2(B - C_2),$$

which, when evaluated at $t = 0$ and $t = 1$, is identical to the slope of the lines joining A and C_1 and joining B and C_2 , respectively. Thus the curve is tangent to these two lines. In addition, for $0 \leq t \leq 1$, the coefficients of $B(t)$ are all nonnegative and sum to one, and hence $B(t)$ is in the convex hull of $\{A, C_1, C_2, B\}$.

Bézier cubics are flexible enough to cover the needs of almost every situation. Here is a strategy for constructing such curves:

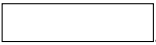
- Choose the starting and ending points of the curve.
- Select lines through those points which are to be tangent to the curve.
- Pick one control point on each line. Being near the starting and ending point will give a flat curve. Moving the control points further away from the starting and ending points will increase the curvature.

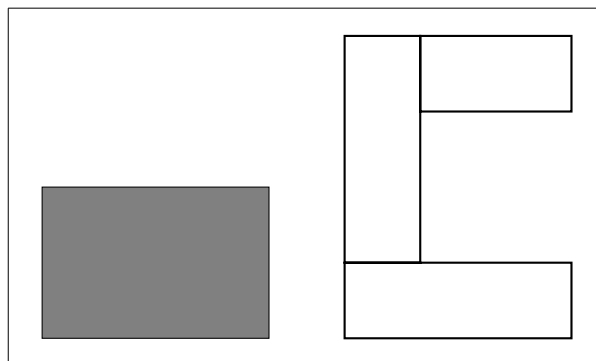
14.1.5 A few more predefined objects

Rectangles

A rectangle is defined using the coordinates of the lower left and upper right corners.

`\draw(0,0) rectangle (2,1/2);`

results in .



Rectangles

```


\coordinate (A) at (0,0);
\coordinate (B) at (3,2);
\coordinate (C) at (7,0);
\coordinate (D) at (4,1);
\coordinate (E) at (5,4);
\coordinate (F) at (7,3);
\filldraw [fill=gray] (A) rectangle (B);
\draw (C) [thick] rectangle (D) rectangle (E) rectangle (F);

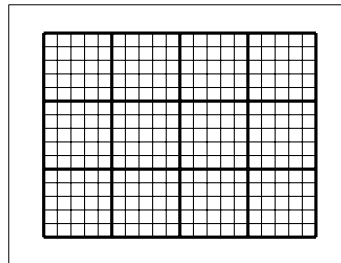
```

Grids

The syntax for grids is similar to that for rectangles. The lower left and upper right corners are parameters. By default the grid squares will have unit sides, but this may be changed setting the parameter `step` to a different dimension. Here is an example:

```
\draw[step=3pt] (0,0) grid (21pt,12pt);
```

results in . Here is another example that simulates graph paper.



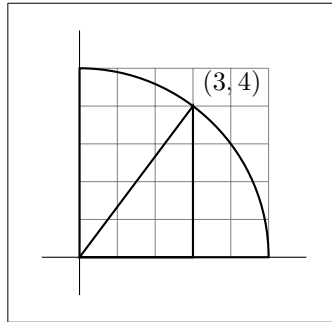
Graph paper

```

\coordinate (A) at (0,0);
\coordinate (B) at (4,3);
\draw [very thick] (A) grid (B);
\draw [step=1/5] (A) grid (B);

```

There is a specially defined grid style [`help lines`] that is useful while creating TikZ figures. It draws a background grid with light gray lines which may be used to see exactly where objects appear on the page.



Help lines

```

\coordinate [label=above right:{$(3,4)$}] (A) at (3,4);
\draw (-1,0)--(6,0) (0,-1)--(0,6); % x-axis and y-axis
\draw [help lines] (0,0) grid (5,5);
\draw [thick] (0,0)--(3,0)--(3,4)--cycle;
\draw [thick] (5,0) arc [start angle=0, end angle=90, radius=5]
-- (0,0) -- cycle;

```

14.2 Some tools of the trade

TikZ provides many tools to modify and simplify the process of putting objects on the page. This section describes some of these tools; they greatly improve productivity.

14.2.1 Transformations

Global modifications versus local modifications

Global modifications apply to the entire TikZ figure; local modifications apply only to a part of it.

To make a global modification, add a parameter to the TikZ environment. For example: `scale=1.2` will enlarge the figure by 20%:

```

\begin{tikzpicture}[scale=1.2]
...
\end{tikzpicture}.

```

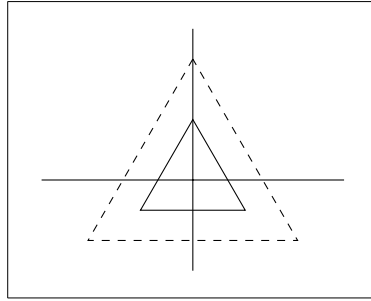
To make this modification locally, use the predefined environment `scope`.

```

\begin{tikzpicture}
...
\begin{scope}[scale=1.2]
...
\end{scope}
...
\end{tikzpicture}.

```

The enlargement will only take place within the scope environment. In the following example, only the dashed triangle is enlarged.



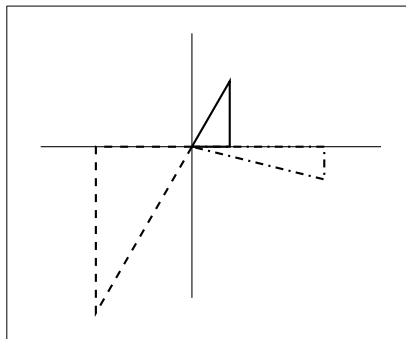
Scope example

```
\draw (-2.5,0)--(2.5,0) (0,-1.5)--(0,2.5); % x-axis and y-axis
\newcommand\mytriangle[1]{\draw[#1] (90:1)--(210:1)--(330:1)--cycle;}
\mytriangle{solid}
\begin{scope}[scale=2.0]
\mytriangle{dashed}
\end{scope}
```

Scale

The parameter `scale(t)`, in effect, multiplies the x and y coordinates by t . This enlarges the figure by a factor of t (which may be less than 1 or negative). So for example, to change the unit distance from the default of 1cm to 1in, use `scale(2.54)`.

In addition, the x -axis and y -axis may be scaled separately by setting the parameters `xscale` and `yscale`.



Scales applied to a right triangle

```
\draw (-2.0,0)--(2.5,0) (0,-2.0)--(0,1.5); % x-axis and y-axis
\newcommand{\RightTriangle}[1]{
\coordinate (A) at (0,0); \coordinate (B) at (0.5,0);
\coordinate (C) at (60:1);
```



```

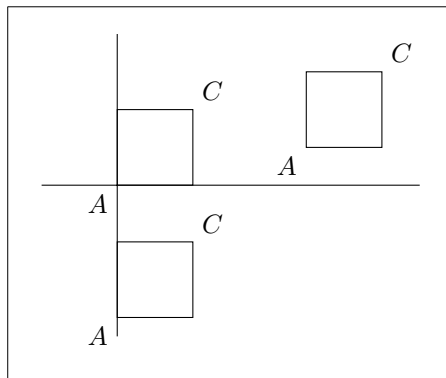
\draw [#1] (A)--(B)--(C)--cycle;}
\RightTriangle{thick}
\begin{scope} [scale=-2.54]
\RightTriangle{thick, dashed}
\end{scope}
\begin{scope} [xscale=3.5, yscale=-0.5]
\RightTriangle{thick, dashdotted}
\end{scope}

```

Note that the type size for any text or labels is unchanged by rescaling.

Shift

The parameter `shift(x,y)` adds (x,y) to each coordinate. The effect is to translate the entire figure by a factor of (x,y) . In a similar manner, shifting in the direction of the x -axis or y -axis may be done using `xshift` and `yshift`.



Shifts applied to the unit square

```

\draw (-1,0)--(4,0) (0,-2)--(0,2); % x-axis and y-axis
\newcommand{\unitsquare}{
  \coordinate [label=below left:$A$] (A) at (0,0);
  \coordinate (B) at (1,0);
  \coordinate [label=above right:$C$] (C) at (1,1);
  \coordinate (D) at (0,1);
  \draw (A)--(B)--(C)--(D)--cycle;}
\unitsquare
\begin{scope} [shift={(5/2,1/2)}] \unitsquare \end{scope}
\begin{scope} [yshift=-1.75cm] \unitsquare \end{scope}

```

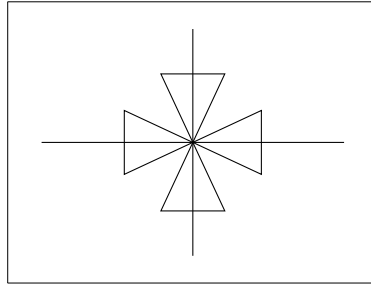


It figures ...

- Note the braces in `[shift={(2,1/2)}]`. They are necessary: remember that the parameters form a comma-separated list, so without the braces the comma will be interpreted as a parameter separator rather than as a part of the notation for a vector.
-

Rotate

The parameter `[rotate= θ]` will cause the material within the `scope` environment to be rotated around the origin by θ° .

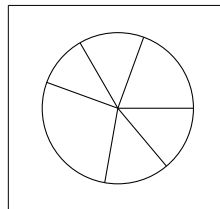


Triangle rotation around the origin

```
\draw (-2.0,0)--(2.0,0) (0,-1.5)--(0,1.5); % x-axis and y-axis
\newcommand{\DrawVane}{\draw (0,0) -- (25:1) -- (-25:1) -- cycle;}
\DrawVane
\begin{scope} [rotate=90] \DrawVane \end{scope}
\begin{scope} [rotate=180] \DrawVane \end{scope}
\begin{scope} [rotate=270] \DrawVane \end{scope}
```

14.2.2 A little repetition

For this section the viewpoint changes: the goal is to put objects on the page more efficiently. The tool used is `\foreach`. The next example uses it to make a little pie chart:



A pie chart

```
\foreach \k in {0,70,120,160,260,310} {
  \draw (0,0)--(\k:1);
}
\draw (0,0) circle [radius=1];
```

Here's how the syntax is interpreted:

- `\foreach \k in {0,70,120,160,260,310}`: `\k` acts like a variable. It is successively assigned the values that lie within the first pair of braces, that is, 0, 70, 120, and so forth.
- After `\k` is assigned its value, it is substituted into the expression that follows within the second set of braces. In this case, it means that it executes the commands sequentially:

```
\draw{(0,0)--(0:1);}
\draw{(0,0)--(70:1);}
\draw{(0,0)--(120:1)}
\vdots
\draw{(0,0)--(310:1)}.
```

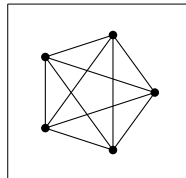
Upon completion of the sequence, the `\foreach` command is finished.

- The circle is then drawn to complete the pie chart, and the figure is completed.

There are some shortcuts available when using `\foreach`:

- If `\foreach` executes only one command, the second set of braces is not necessary. In the previous example
`\foreach \k in {0,70,120,160,260,310} \draw (0,0)--(\k:1);`
 is equivalent to the three lines it replaces.
- If i and j are integers, then $\{i, \dots, j\}$ is the same as the set of integers from i to j , inclusive.
- If i , j , and k are integers, then $\{i, k \dots, j\}$ is the same as the arithmetic sequence of integers from i to j with common difference $d = k - i$, that is, the sequence $i, i + d, i + 2d, \dots$
- The last row of Table 14.5 contains a message: beware of round off errors when using nonintegral entries. In particular, changing the last entry to $\{1, 1.3, \dots, 1.9\}$ will not give the expected result.

The following example uses `\foreach` to draw the complete graph K_n .



The graph K_5

```

\renewcommand\mydot[1]{\fill #1 circle [radius=2pt]}
\newcommand\vert{5}
\foreach \i in {1,...,\vert} {
  \coordinate (P\i) at (360/\vert*\i:1);
  \mydot{(P\i)};
  \foreach \j in {1,...,\i} \draw (P\i)--(P\j);
}

```



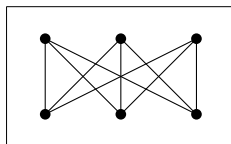
It figures...

- Compare this code with that of C_5 on page 364 that doesn't use `\foreach`. The new code is much more compact.
- You may also use this example for complete graphs of any size by changing the defined value of `\vert`.
- TikZ can do some limited computations involving multiplication and division when defining coordinates.

Entered version	Expanded version
$\{1, \dots, 6\}$	$\{1, 2, 3, 4, 5, 6\}$
$\{6, \dots, 1\}$	$\{6, 5, 4, 3, 2, 1\}$
$\{1, 3, \dots, 11\}$	$\{1, 3, 5, 7, 9, 11\}$
$\{1, 1.3, \dots, 2\}$	$\{1, 1.3, 1.6, 1.90001\}$

Table 14.5: Shortened foreach parameters

It is also possible to nest `\foreach` statements. The next example shows how to draw the complete bipartite graph $K_{m,n}$ using nested `\foreach` statements. (The graph $K_{m,n}$ has $m + n$ vertices with each of the first m being joined by an edge to each of the last n .)



The complete bipartite graph $K_{3,3}$

```

\newcommand\mvert{3} \newcommand\vert{3}
\foreach \i in {1,...,\mvert} {
  \coordinate (P\i) at (\i,0);

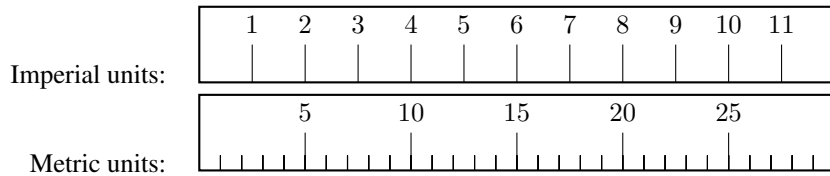
```

```

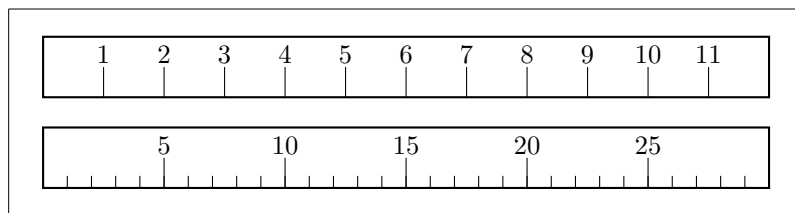
\fill (P\i) circle [radius=2pt];
}
\foreach \j in {1,...,\nvert} {
  \coordinate (Q\j) at (\j,1);
  \fill (Q\j) circle [radius=2pt];
}
\foreach \i in {1,...,\mvert}
  \foreach \j in {1,...,\nvert}
    \draw (P\i)--(Q\j);

```

The next example is a ruler with a given number of units marked on the lower edge. The general pattern should be something like



It's clear that some internal arithmetic is necessary to get this. In the Imperial case, the ruler is 12 units wide, but there are only 11 labels. The metric case is further complicated in its layout. \LaTeX counters are just what is needed since they support arithmetic operations. See page 443 for the details on the creation and use of counters.



Imperial and metric rulers

```

\newcounter{mycounter}           % create a new counter
\setcounter{mycounter}{11}       % assign a value to the counter
\newcounter{rcounter}           % create a new counter
\setcounter{rcounter}{5}
\foreach \k in {1,...,\themycounter} {
  \draw node [below] at (\k,1) {$\k$} (\k,0)--(\k,1/2);
}
\stepcounter{mycounter}
\draw [thick] (0,0) rectangle (\themycounter,1);

\setcounter{mycounter}{5}
\begin{scope} [yshift=-1.5cm, xscale=2]
\foreach \k in {1,...,\themycounter} {

```

```

\draw node [below] at (\k,1) {\$\thercounter\$} (\k,0)--(\k,1/2);
\addtocounter{rcounter}{5}
}
\stepcounter{mycounter}
\foreach \k in {0.2,0.4,...,\themycounter} \draw (\k,0)--(\k,1/5);
\draw [thick] (0,0) rectangle (\themycounter,1);
\end{scope}

```

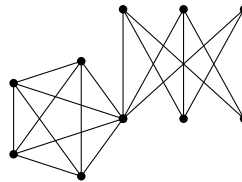


It figures...

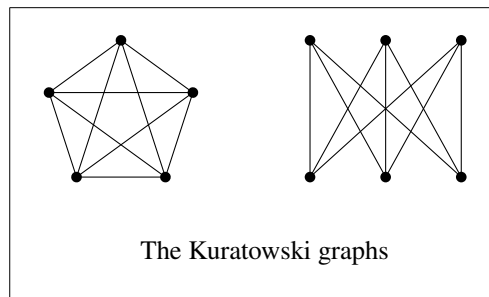
- `\themycounter` is the current value of the counter `mycounter`.
- The same technique works for marking the x -axis and the y -axis of a graph.

Cut and paste: `\scope` is your friend

The graphs of the complete bipartite graph K_5 on page 375 and the complete bipartite graph $K_{3,3}$ on page 376 have been displayed using TikZ. We wish to display them together. As a start, simply cut and paste the code for each and put them into a single file. The result in this case:



Not ideal! Among other things, it is necessary move the graphs so that they don't overlap and they need to be mutually aligned.. The fix is really easy: put each graph within its own scope environment and use the parameters `shift`, `rotate`, and `scale` to move, rotate, and resize the two graphs until they are balanced and attractive.



```

\renewcommand\mydot[1]{\fill #1 circle [radius=2pt]}

\begin{scope}[shift={(-1.5,\sin(54))}, rotate=18]

```

```

\newcommand\nvert{5}
\foreach \i in {1,...,\nvert} {
  \coordinate (P\i) at (360/\nvert*\i:1);
  \mydot{(P\i)};
  \foreach \j in {1,...,\i} \draw (P\i)--(P\j);
}
\end{scope}

\begin{scope}
  \newcommand\mvert{3}
  \newcommand\nvert{3}
  \foreach \i in {1,...,\mvert} {
    \coordinate (P\i) at (\i,0);
    \mydot{(P\i)};
  }
  \foreach \j in {1,...,\nvert} {
    \coordinate (Q\j) at (\j,{1+sin(54)});
    \mydot{(Q\j)};
  }
  \foreach \i in {1,...,\mvert}
    \foreach \j in {1,...,\nvert}
      \draw (P\i)--(Q\j);
\end{scope}
\node at (0.4,-1) {The Kuratowski graphs};

```

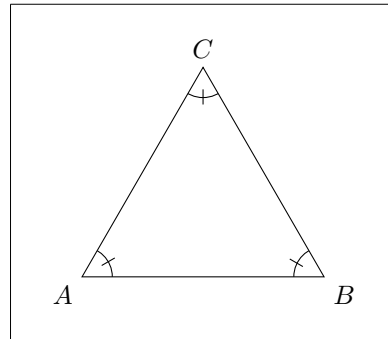


It figures...

- Notice that we define `\nvert` twice using `\newcommand`. This normally is an error that will cause \LaTeX to quit. However, what starts in scope stays in scope, and any new definition disappears when we exit the environment.
 - We apply the parameters in left-to-right order. Order matters!
 - It is easy to include text.
-

Getting clipped

The `\clip` command applies to a closed path. Once invoked, drawing only occurs within the clipped area. In the following example of an equilateral triangle, the arcs at each vertex are initially drawn as a circle and then the part outside of the triangle is clipped. This is surely easier than finding the right coordinates for the `\arc` command.



Equilateral triangle

```

\coordinate [label=below left:$A$] (A) at (0,0);
\coordinate [label=below right:$B$] (B) at (4,0);
\coordinate [label=above:$C$] (C) at (60:4);
\draw (A)--(B)--(C)--cycle;
\clip (A)--(B)--(C)--cycle;
\foreach \ctr in {A,B,C} \draw (\ctr) circle [radius=1/2];
\draw (30:3/8)-- +(30:1/4) (2,3.1) -- +(0,-1/4)
(3.65,0.17) -- +(150:1/4);

```




It figures...

- The order of the code is important. As written, nothing will appear outside the clipping path once `\clip` is invoked. However, if needed, we may include the clipping within a scope environment to limit its application.
- If the clipping path is also being drawn (as in this example), do the drawing first. Only half of the line will be inside the clipping path, and, even worse, computer round off errors may make a hash of things. There is also a *TikZ* alternative for this situation: do both the drawing and clipping using `\clip [draw] (A)--(B)--(C)--cycle;`

14.2.3 Adding some color


Putting color on the page with *TikZ* is usually easy and only involves adding the color name to the parameter list. For example

```
\draw[blue,fill=yellow,scale=4.0] (0,1/15)circle[radius=2pt];
```

results in  a circle filled with yellow with a blue edge.

For a gray arrow:

```
\draw[->,gray] (0,1/10)--(1,1/10);
```

results in .

There are several ways to define a color name:

- Use one of the predefined names:

























Color name		Color name	
red		black	
orange		white	
yellow		gray	
green		lightgray	
blue		darkgray	
indigo		cyan	
violet		brown	
lime		pink	
magenta		purple	
olive		teal	
orange		blue!45!orange	
UManitobaBrown		UManitobaGold	

Table 14.6: Color samples for TikZ

blackbluebrowncyandarkgraygraygreen

lightgraylimemagentaoливorangepink

purpleredtealvioletwhiteyellow

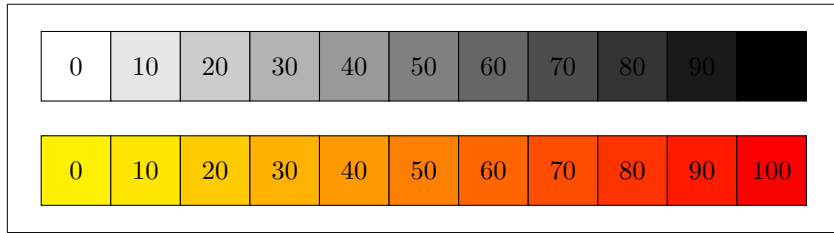
- Define a name using rgb values: the syntax for defining new colors UManitobaBrown and UManitobaGold is

```
\definecolor{UManitobaBrown}{RGB}{79,44,29}
\definecolor{UManitobaGold}{RGB}{242,169,0}.
```

- Mix previously defined names: The color red!30!yellow is a weighted average of their (rgb) values: 30% red and 70% yellow. There is a shortcut if the second color is white: you may omit the name. So color black!60!white is the same as black!60!.

See Table 14.6 for color samples.

The following example shows how to mix colors to make a gray scale or to display a gradual change from one color to another. See the next page in color in the sample file 6colorpages.



Shades of gray and colors

```

\foreach \k in {0,10,...,100} {
  \filldraw[scale=0.1,fill=red!\k!yellow] (\k,0) rectangle +(10,10);
  \filldraw[scale=0.1,fill=black!\k!] (\k,15) rectangle +(10,10);
}

\newcounter{Mycounter} \setcounter{Mycounter}{0}
\foreach \k in {0.5,1.5,...,11} {
  \node (A) at (\k,2) {$\theMycounter$};
  \node (B) at (\k,0.5) {$\theMycounter$};
  \addtocounter{Mycounter}{10}
}

```

14.3 Next Step: text and labels on the page

14.3.1 Labels, labels everywhere

We often put text in an illustration. In particular, we need labels for clarity. In previous examples, we used the `\coordinate` command to add labels. This command is a special case of the keyword node.

The most basic use of the node keyword is within a path. In the path:

```
\draw... (a,b)--(c,d) ...;
```

insert a node:

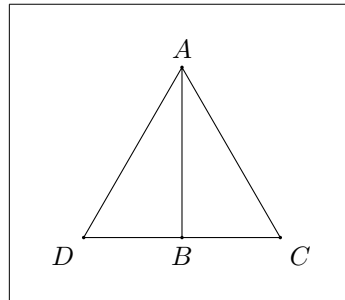
```
\draw... (a,b) node {some text} --(c,d) ... ;
```

Then ‘some text’ prints at the current point in the path, in this case, (a,b) .

Labels are crucial for clarity. Normally a label is a small piece of text placed nearby a given point (called the ‘anchor point’) in the diagram. The exact position of the label is controlled by adding an anchor as a parameter. The anchors are

above	below	left	right
above left	above right	below left	below right

The names indicate where the label will be placed. The next example illustrates the anchors.



An equilateral triangle

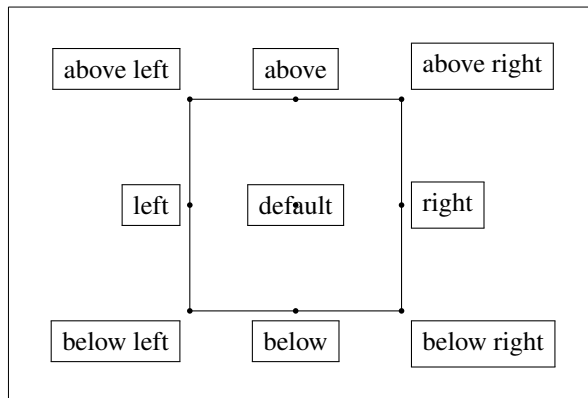
```

\coordinate (A) at (60:4); \coordinate (B) at (2,0);
\coordinate (C) at (4,0); \coordinate (D) at (0,0);
\foreach \pt in {A,B,C,D} \fill (\pt) circle [radius=1pt];

% Now we add nodes to the path (A)--(B)--(C)--(A)--(D)--(B).
\draw (A) node [above] {$A$} --
      (B) node [below] {$B$} --
      (C) node [below right] {$C$} -- (A) --
      (D) node [below left] {$D$} -- (B);

```

The following example places boxed text labels near their anchor points. This is achieved by drawing a cycle with nodes inserted to make the labels.



Labels with different anchors

```

\draw (-2,-2) node [below left] {\fbox{below left}} --
      (0,-2) node [below] {\fbox{below}} --
      (2,-2) node [below right] {\fbox{below right}} --
      (2,0) node [right] {\fbox{right}} --
      (2,2) node [above right] {\fbox{above right}} --
      (0,2) node [above] {\fbox{above}} --
      (-2,2) node [above left] {\fbox{above left}} --
      (-2,0) node [left] {\fbox{left}} --
      (-2,-2) node [below left] {\fbox{below left}};

```

```

(-2,2) node [above left] {\fbox{above left}} --
(-2,0) node [left] {\fbox{left}} --
cycle;
\foreach \i in {-2,0,2}
  \foreach \j in {-2,0,2}
    \fill (\i,\j) circle [radius=1.5pt];
\draw (0,0) node {\fbox{default}};

```

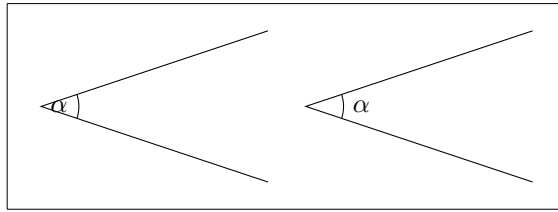


It figures...

- The last line has a path that does nothing but set the current point to (0,0). There is an alternative and perhaps clearer syntax that does the same:

```
\node at (0,0) {\fbox{default}}}
```

Sometimes the labels still don't end up in the right place. The next example changes the distance from the anchor point to the label by giving an explicit dimension to the anchor parameter.



Moving the label

```

\begin{scope}
\foreach \ptname/\i/\j in {A/3/2, B/0/1, C/3/0}
  \coordinate (\ptname) at (\i,\j);
\draw (A)--(B)--(C);
\node [right] at (B) {\alpha};
\clip (A)--(B)--(C)--cycle;
\draw (B) circle [radius=0.5cm];
\end{scope}

\begin{scope}[xshift=3.5cm]
\foreach \ptname/\i/\j in {A/3/2, B/0/1, C/3/0}
  \coordinate (\ptname) at (\i,\j);
\draw (A)--(B)--(C);
\node [right=0.5cm] at (B) {\alpha};
\clip (A)--(B)--(C)--cycle;
\draw (B) circle [radius=0.5cm];
\end{scope}

```



It figures...

- The left figure uses the default value for node `[right]`, while the right figure uses node parameter `[right=0.5cm]` to move the label.

- The `\foreach` command can assign values to more than one variable by using `/` as a separator. The first assignment of the `\foreach` statement sets `\ptname`, `\i`, and `\j` to A, 3, and 2 respectively.

Special positioning of labels

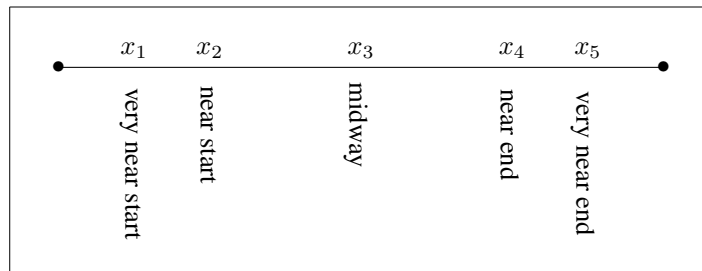
Sometimes a label is used to identify a line or a curve; the single label is often put somewhere near the middle, or two labels are put somewhere near the ends. This may be done easily by adding a new parameter to the node:

```

very near start    near start
                  midway
near end          very near end

```

The next example shows these positions along a straight line.



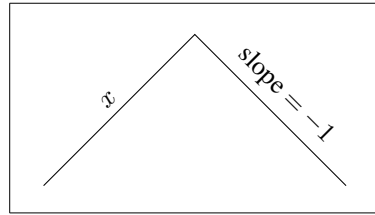
Special positions of labels

```

\draw (0,5) node {$\bullet$} -- (8,5) node {$\bullet$}
node [above, very near start] {$x_1$}
node [below, very near start, rotate=270, xshift=0.5in, yshift=6pt]
{very near start}
node [above, near start] {$x_2$}
node [below, near start, rotate=270, xshift=0.35in, yshift=6pt]
{near start}
node [above, midway] {$x_3$}
node [below, midway, rotate=270, xshift=0.3in, yshift=6pt] {midway}
node [above, near end] {$x_4$}
node [below, near end, rotate=270, xshift=0.35in, yshift=6pt]
{near end}
node [above, very near end] {$x_5$}
node [below, very near end, rotate=270, xshift=0.5in, yshift=6pt]
{very near end};

```

Sometimes you may want to rotate the label so that it has the same slope as the curve it is labelling. The parameter `sloped`, often used in conjunction with `midway`, does just that.



Sloped labels

```
\draw (2,0)--(4,2) node [above, sloped, midway] {$x$} --
      (6,0) node [above, sloped, midway]{slope $= -1$};
```

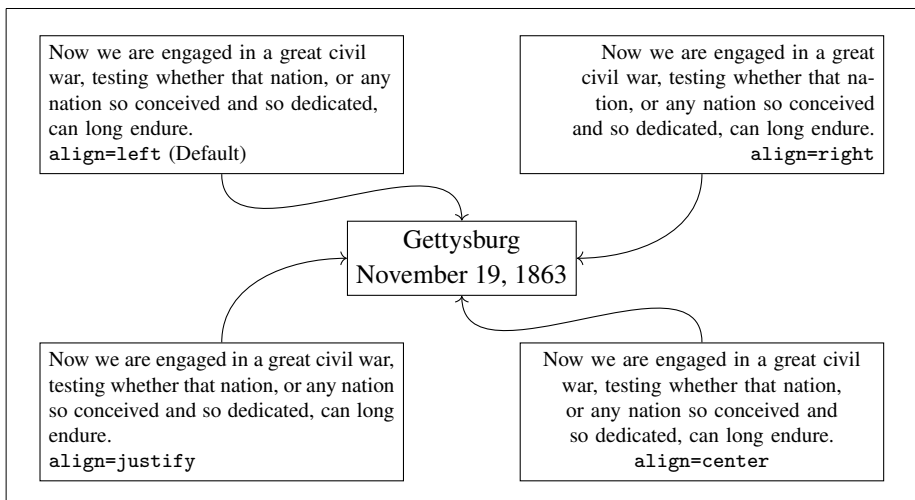
14.3.2 Larger blocks of text

If a TikZ figure needs a large block of formatted text, put the text in a node. By default the text goes on one (possibly very long) line. To format the text:

- Use a double backslash `\\` to insert a line break, or
- Set the parameter `text width` to the desired width of the text. In this case, also set the `align` parameter: to

```
align=left      align=right
align=center    align=justify
```

The next example shows how the same text in a diagram may be formatted in several different ways.



Text diagrams

```

\newcommand\gettysburg{
Now we are engaged in a great civil war, testing whether that
nation, or any nation so conceived and so dedicated, can long
endure.}
\node (A) [draw, font=\footnotesize, text width=1.8in]
  at (0,0.8in) {\gettysburg\ \texttt{align=left} (Default)};
\node (B) [draw, font=\footnotesize, text width=1.8in, align=right]
  at (2.5in,0.8in) {\gettysburg\ \texttt{align=right}};
\node (C) [draw, font=\footnotesize, text width=1.8in, align=center]
  at (2.5in,-0.8in) {\gettysburg\ \texttt{align=center}};
\node (D) [draw, font=\footnotesize, text width=1.8in, align=justify]
  at (0,-0.8in) {\gettysburg\ \texttt{align=justify}} ;
\node (E) at (1.5in,0in){};
\node (E) [draw, align=center] at (1.25in,0in)
  {\gettysburg\ November 19, 1863};

\draw [->] (A) to [out=270, in=90] (E);
\draw [->] (B) to [out=270, in=0] (E);
\draw [->] (C) to [out=90, in=270] (E);
\draw [->] (D) to [out=90, in=180] (E);

```

14.3.3 Let TikZ do the computations

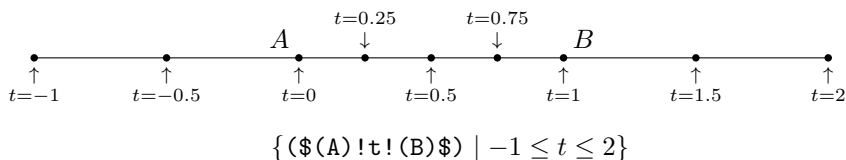
TikZ can do many computations. Usually this requires loading the library:

```
\usetikzlibrary{calc}
```

Some computations are used frequently enough to acquire their own notation. The following subsections give some that are particularly useful.

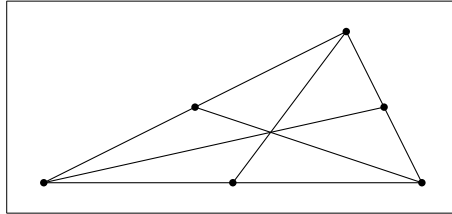
Points along a line

Given two points A and B , it may be useful to identify the points on the line joining them. The TikZ notation for this is $(A)!\texttt{t}!(B)$, where t is a real number. For $t = 0$, the point is A , for $t = 1$, the point is B , and as t ranges from 0 to 1, the points go from A to B , which may be described by $(A)!\texttt{t}!(B) = (1 - t)A + tB$. Here is a display of the points for some different values of t .



This special notation make sense: the outer parentheses indicate that a coordinate is being defined, the dollar signs indicate that a mathematical computation is taking place, and the inner part shows what is being evaluated. For example, use $(A)!0.5!(B)$ for the coordinates of the midpoint of the line joining A and B .

The next example takes a triangle and joins each vertex to the midpoint of the opposite side.

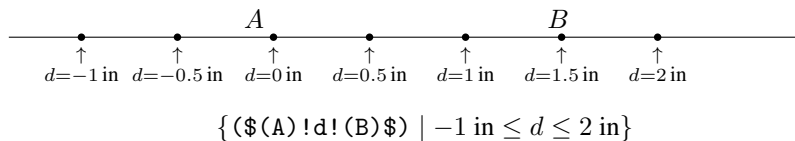


Vertex to opposite line bisector

```
% uses: \usetikzlibrary{calc}
\coordinate (A) at (0,0); \coordinate (B) at (5,0);
\coordinate (C) at (4,2); \coordinate (A') at ($(B)!0.5!(C)$);
\coordinate (B') at ($(A)!0.5!(C)$);
\coordinate (C') at ($(A)!0.5!(B)$);
\draw (A) --(B)--(C)--cycle;
\foreach \pt in {A,B,C} {
  \draw (\pt)--(\pt');
  \fill (\pt) circle [radius=1.4pt];
  \fill (\pt') circle [radius=1.4pt];
}
```

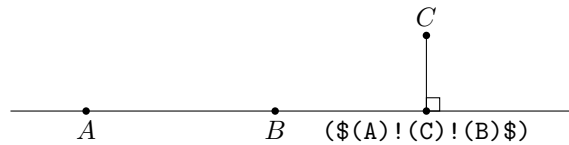
The real number t can be replaced by a dimension d . In this case $(A)!d!(B)$ is the point on the line joining A and B at distance $|d|$ from A (if $d > 0$, the direction is from A to B).

Here is a sample of points:

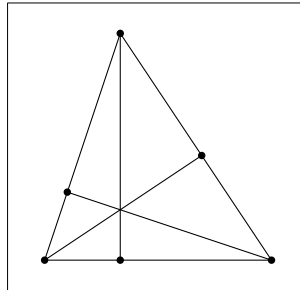


Dropping a perpendicular

Definitions of two specific points on a line joining A and B are given in the previous section, namely, $(A)!t!(B)$ where t is a real number and $(A)!d!(B)$ where d is a dimension. We define another point on the same line: $(A)!(C)!(B)$ where (C) is a coordinate. It is the point obtained by dropping a perpendicular from C to the line joining A and B .



The next example uses this new definition to drop perpendiculars from each vertex of a triangle to the opposite side.



Perpendiculars

```
% uses: \usetikzlibrary{calc}
\coordinate (A) at (0,0); \coordinate (B) at (3,0);
\coordinate (C) at (1,3); \coordinate (A') at ($ (B) ! (A) ! (C) $);
\coordinate (B') at ($ (A) ! (B) ! (C) $);
\coordinate (C') at ($ (A) ! (C) ! (B) $);
\draw (A)--(B)--(C)--cycle;
\foreach \pt in {A,B,C} {
  \draw (\pt)--(\pt');
  \fill (\pt) circle [radius=1.4pt];
  \fill (\pt') circle [radius=1.4pt];
}
```


Coordinate computations

Coordinates may be computed using

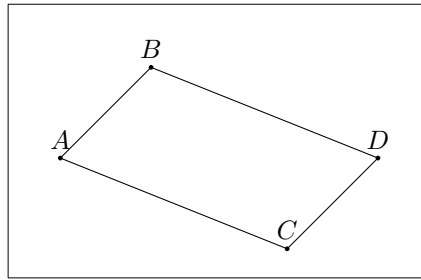
```
\usetikzlibrary{calc}
```

The pattern used in the previous subsections continues: The outer parentheses indicate a coordinate, the dollar signs indicate a computation, the inner part is what is being computed. For example

```
\draw(0,0)--($1/2*(1,0)+1/3*(0,1)$);
```

results in  where the second point is computed as $(\frac{1}{2}, \frac{1}{3})$.

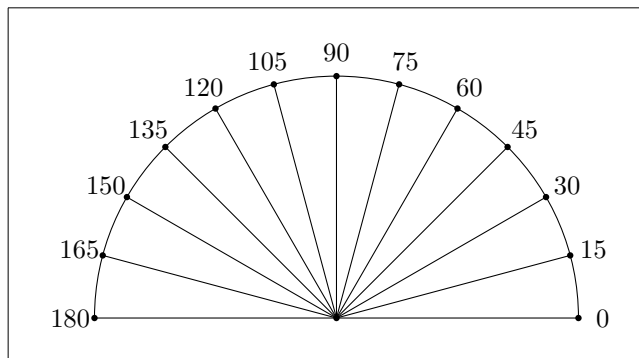
The next example starts with three noncolinear points and completes the parallelogram.



Completing the parallelogram

```
% uses: \usetikzlibrary{calc}
\coordinate (A) at (1,2);
\coordinate (B) at (3,4);
\coordinate (C) at (6,0);
\coordinate (D) at ($(B)+(C)-(A)$);
\draw (A)--(B)--(D)--(C)--cycle;
\foreach \pt in {A,B,C,D} {
  \node [above] at (\pt) {$\pt$};
  \fill (\pt) circle [radius=1.5pt];
}
```

The following example draws a protractor. A small computation in the coordinates gives an elegant radial labelling.



A protractor

```
\newcommand\Radius{4}
\foreach \k in {0,15,...,180} {
  \draw (0,0)--(\k:\Radius);           % draw the radius
  \node at (\k:{1.1*\Radius}) {$\k$}; % print the number
  \fill (\k:\Radius) circle [radius=1.5pt]; % add a dot
}
\fill (0,0) circle [radius=1.5pt];
\draw (\Radius,0) arc [start angle=0, end angle=180, radius=\Radius];
```

Intersection points

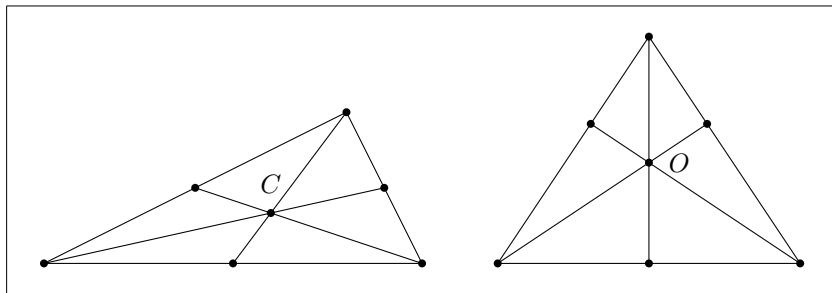
One helpful computation available with TikZ is the finding of intersection points between curves. This is done with a special library invoked by

```
\usetikzlibrary{intersections}
```

To find the intersection point of two lines: if (A) and (B) are two points on the first line, (C) and (D) are on the second line, then the intersection point (E) is determined by:

```
\coordinate (E) at (intersection of A--B and C--D);
```

We already have two examples of triangles with intersecting lines. In both cases, a dot and a label can be added at the common intersection point of the three lines.



Compute line intersections

```
% uses: \usetikzlibrary{calc,intersections}
\coordinate (A) at (0,0); \coordinate (B) at (5,0);
\coordinate (C) at (4,2); \coordinate (A') at ($(B)!0.5!(C)$);
\coordinate (B') at ($(A)!0.5!(C)$);
\coordinate (C') at ($(A)!0.5!(B)$);
\draw (A) --(B)--(C)--cycle;
\foreach \pt in {A,B,C} {
  \draw (\pt)--(\pt');
  \fill (\pt) circle [radius=1.5pt];
  \fill (\pt') circle [radius=1.5pt];
}
\coordinate (E) at (intersection of A--A' and B--B');
\fill (E) circle [radius=1.5pt];
\node [above=4pt] at (E) {$C$};

\begin{scope} [xshift=7cm]
\coordinate (A) at (-1,0); \coordinate (B) at (3,0);
\coordinate (C) at (1,3); \coordinate (A') at ($(B)!(A)!(C)$);
\coordinate (B') at ($(A)!(B)!(C)$);
\coordinate (C') at ($(A)!(C)!(B)$);
\draw (A) --(B)--(C)--cycle;
\foreach \pt in {A,B,C} {
  \draw (\pt)--(\pt');
  \fill (\pt) circle [radius=1.5pt];
}
```

```

\fill (\pt') circle [radius=1.5pt];
}
\coordinate (E) at (intersection of A--A' and B--B');
\fill (E) circle [radius=1.5pt];
\node [right=4pt] at (E) {$0$};
\end{scope}

```



It figures...

- `\coordinate` is equivalent to `\draw [coordinate]`

Finding the intersection points for more general curves is a little more difficult since several somewhat more complicated parameters must be used.

- We must name each curve using the parameter name `path`:

```

\draw [name path=ellipse] (0,0) ellipse [x radius=2,
y radius=1];

```

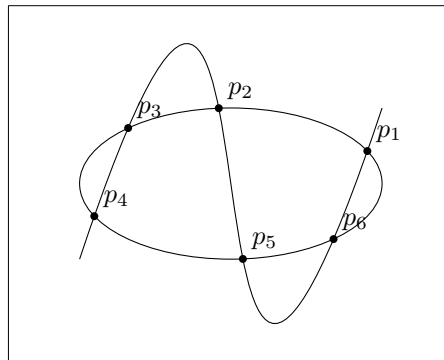
```

\draw [name path=cubic] (-2,-1)..controls (1,8) and
(-1,-8)..(2,1);

```

- `[name intersections={of=cubic and ellipse, name=p, total=\t}]` takes the two curves named `cubic` and `ellipse`, computes the intersection points (`p-1`), (`p-2`), etc., the total number of such points being `\t`.

The next example puts these pieces together.



Compute curve intersections

```

% uses \usetikzlibrary {intersections}
\clip (-2.5,-2) rectangle (2.5,2);          % crop out control points
\draw [name path=cubic] (-2,-1)..controls (1,8) and (-1,-8)..(2,1);
\draw [name path=ellipse] (0,0) ellipse [x radius=2, y radius=1];

\fill [name intersections={of=cubic and ellipse, name=p, total=\t}]
\foreach \i in {1,...,\t} {

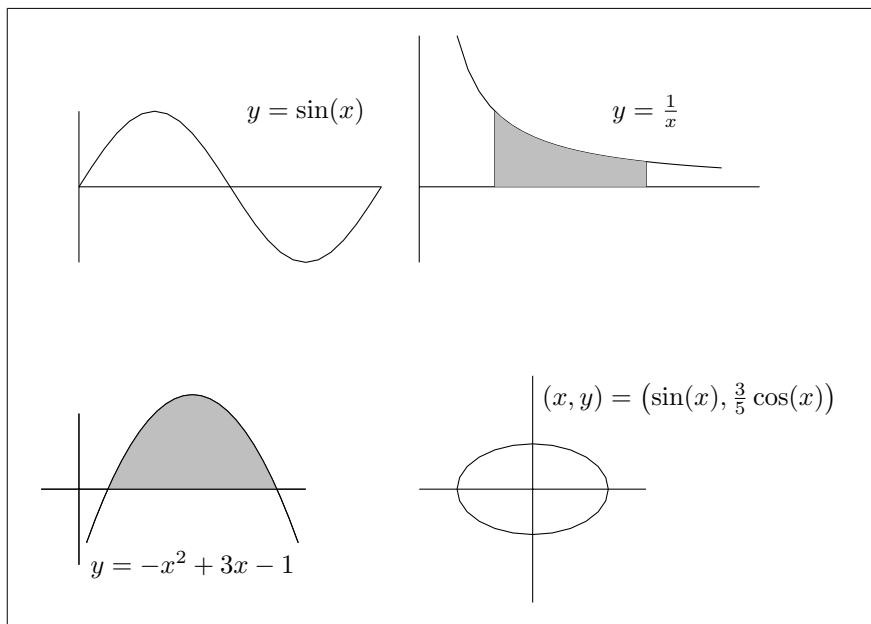
```

```
(p-\i) circle [radius=1.5pt]
node [above right] {$p\_i$}
};
```

14.4 Graphs of functions and presentations of data

Drawing plots of functions and creating presentations of data is a complex topic whose discussion in depth goes beyond the scope of this introduction. Still, simple plots and presentations are possible with the tools so far developed. A simple pie chart is easy. Similarly, it is not difficult to label axes. We can create a simple horizontal or vertical bar chart by filling rectangles attached to the axes.

Simple plots of functions are not difficult either. The pattern for doing so may be seen in the following samples.



Plots

```
\begin{scope}
\draw (0,0)--(4,0) (0,-1)--(0,1);
\draw [domain=0:360, xscale=1/90] plot (\x,{sin(\x)});
\node at (3,1) {$y=\sin(x)$};
\end{scope}

\begin{scope} [xshift=4.5cm]
\draw (0,0)--(4.5,0) (0,-1)--(0,2);
\draw [domain=0.5:4] plot (\x,1/\x);
\draw (1,0)--(1,1) (3,0)--(3,1/3);
```

```

\mode at (3,1) {\$y=\frac{1}{x}};
\fill [domain=1:3, lightgray, variable=\x]
  (1,1)--plot (\x,1/\x) -- (3,0) -- (1,0) -- cycle;
\end{scope}

\begin{scope} [yshift=-4cm]
\draw [name path=axes] (-0.5,0)--(3,0) (0,-1)--(0,1);
\draw [name path=parabola, domain=0.1:2.9] plot (\x,{-\x^2+3*\x-1});
\fill [lightgray, name intersections={of=parabola and axes, name=p}]
  (p-1)--
  plot [domain=((3-sqrt(5))/2:(3+sqrt(5))/2)] (\x,{-\x^2+3*\x-1}) --
  (p-2)--cycle;
\mode at (1.5,-1) {\$y=-x^2+3x-1};
\draw [domain=0.1:2.9] plot (\x,{-\x^2+3*\x-1}); %get full line width
\draw (-0.5,0)--(3,0) (0,-1)--(0,1);
\end{scope}

\begin{scope} [shift={(6cm,-4cm)}]
\draw (-1.5,0)--(1.5,0) (0,-1.5)--(0,1.5);
\draw [domain=0:360] plot({sin(\x)},{0.6*cos(\x)});
\mode [right] at (0,1.2)
  {\$(x,y)=\left(\sin(x),\frac{35}{cos(x)}\right)\$};
\end{scope}

```



It figures ...

- These examples are all done as parametric plots. The fourth example is explicitly parametric, while the other functions are straightforward with the first coordinate being set to the default variable $\backslash x$.
- The default unit for trigonometric functions is the degree. Either converting to radians or setting `xscale` can be used adjust the plot.

TikZ has a library with many features for presenting data:

```
\usetikzlibrary{datavisualization}
```

Its usage is described in great detail in the TikZ and PGF manual available on line at <http://tikz.dev>.

14.5 A little planning goes a long way

We described many any tools that enable the creation of TikZ figures. With more complicated projects careful planning is often useful. The following example and hints give a strategy to make this process easier.

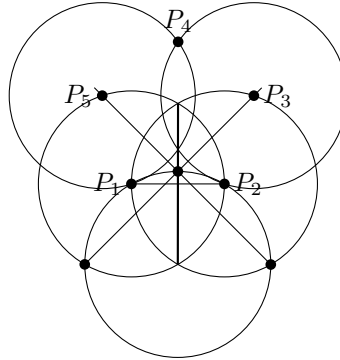
A good general strategy includes:

- Divide and conquer: Build up the figure from the smallest feasible units.
- If the figure has several layers, work from the bottom layer up.

- Labels should be mnemonic.

14.5.1 An interesting example

We want the following picture. We shall build it using some simple steps.



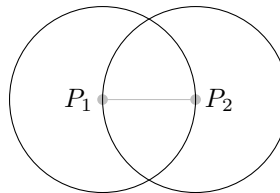
1. Draw an initial line of unit length joining P_1 and P_2 .

```
\newcommand\Unitlength{35pt}
\coordinate [label=left:  $P_1$ ] (P1) at (0,0);
\coordinate [label=right:  $P_2$ ] (P2) at (\Unitlength,0);
\draw (P1)--(P2);
\mydot{(P1)}; \mydot{(P2)};
```



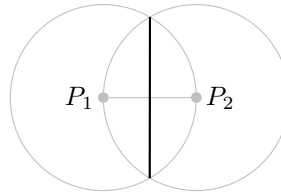
2. Draw unit circles with centers at P_1 and P_2 .

```
\draw [name path=Circle1] (P1) circle [radius=\Unitlength];
\draw [name path=Circle2] (P2) circle [radius=\Unitlength];
```



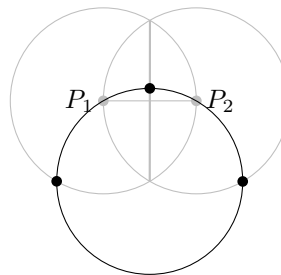
3. Join the intersection points of the circles to construct the perpendicular bisector of the line $\overline{P_1P_2}$.

```
\draw [name intersections={of=Circle1 and Circle2, name=a}]
      (a-1)--(a-2);
\draw [name path=PerpendicularBisector,thick] (a-1)--(a-2);
```



4. Draw a unit circle with center at the lower point of intersection and find the new intersection points.

```
\draw [name path=Circle3] (a-2) circle [radius=\Unitlength];
\node [name intersections={of=Circle3 and
      PerpendicularBisector,name=b}, above right] at (b-1) {};
\mydot{(b-1)};
\node [name intersections={of=Circle3 and Circle2, name=c},
      right] at (c-2) {};
\node [name intersections={of=Circle1 and Circle3, name=d},
      left] at (d-2) {};
\mydot{(c-2)}; \mydot{(d-2)};
```



5. Draw extended lines from the new intersection points through the perpendicular bisector to intersect to first two circles.

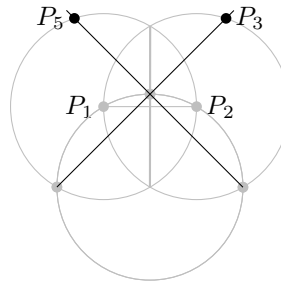
```
\draw [name path=Line1] (d-2)--($ (b-1)!-0.9!(d-2)$);
\draw [name path=Line2] (c-2)--($ (b-1)!-0.9!(c-2)$);
\node [name intersections={of=Circle2 and Line1, name=f},
      right] at (f-1) {$P_3$};
\coordinate (P3) at (f-1);
```



```

\node [name intersections={of=Circle1 and Line2, name=g},
      left] at (g-1) {$P_5$};
\mydot{(f-1)}; \mydot{(g-1)};

```

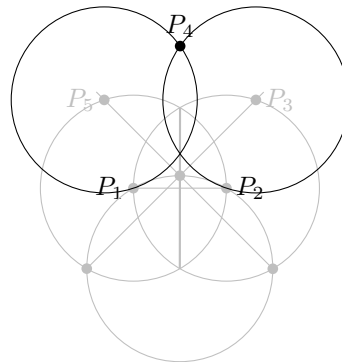


6. Draw unit circles through the new intersection points. The upper intersection point is P_4 .

```

\draw [name path=Circle4] (g-1) circle [radius=\Unitlength];
\draw [name path=Circle5] (f-1) circle [radius=\Unitlength];
\node [name intersections={of=Circle4 and Circle5, name=h},
      above] at (h-1) {$P_4$};
\coordinate (P4) at (h-1);
\coordinate (P5) at (g-1);
\mydot{(h-1)};

```

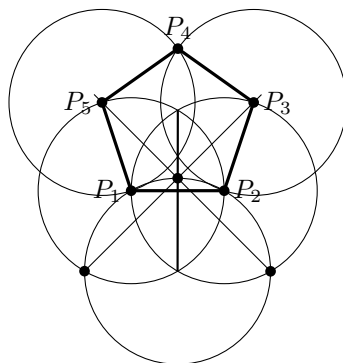


7. A bonus: draw the cycle through the labeled points.

```

\draw [very thick] (P1)--(P2)--(P3)--(P4)--(P5)--cycle;

```



It figures...

- This figure started with two points at unit distance and used the following constructions:
 - Drawing unit circles having defined points as center.
 - Drawing extended lines through two defined points.
 - Finding new defined points by intersecting known circles and lines.

This makes it a ‘compass and straightedge’ construction. The study of objects so constructable, including that of the regular pentagon [27], has a long and interesting mathematical history [41].

14.6 Conclusions

14.6.1 TikZ manual

How much of TikZ did we cover? As compared to the 1200 page manual, obviously not much. Look at the TikZ drawing in Figure 14.1.² There is more to learn...

It seems to us that we covered enough to tackle the simpler requirements of lattice theory and graph theory. We hope that we built the foundation so that you can dig deeper, if necessary, in the big manual.

14.6.2 Vector graphics

Vector graphics software is a type of application used to create, edit, and manage vector images. Unlike raster graphics (like those made in Adobe Photoshop), which use pixels to represent images, vector graphics are made up of paths defined by math equations. These paths are made up of points, lines, curves, and shapes that are based on mathematical expressions, allowing the image to be scaled up or down in size without losing quality.

²See this figure in color in the sample file `6colorpages`.

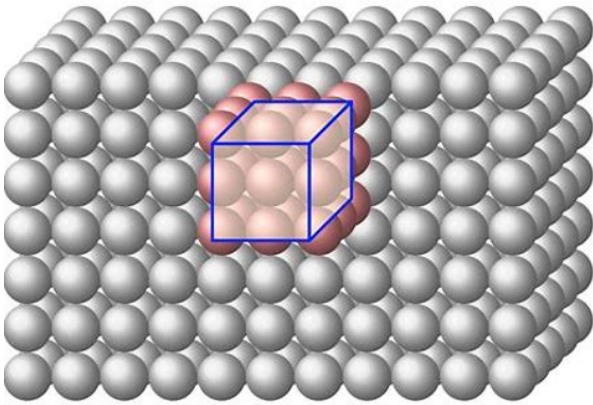


Figure 14.1: A more complicated TikZ drawing

The best known vector graphics application is Adobe Illustrator: A widely used professional vector graphics editor with many features for creating complex vector art and illustrations.

It is easy to learn enough Illustrator to tackle the simple requirements of lattice theory and graph theory, see Figures 14.2 and 14.3. But Illustrator is incredibly powerful, as exemplified by the following two examples on the next page.

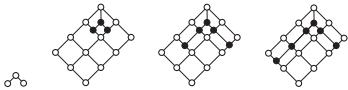


Figure 14.2: Inserting a fork

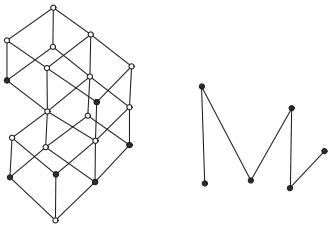


Figure 14.3: A finite distributive lattice D and the ordered set of join-irreducible elements P of D

See the next two figures in color in the sample file 6colorpages.



Figure 14.4: An Illustrator example



Figure 14.5: Another Illustrator example

PART VII

Customization



Commands and environments

Donald E. Knuth designed $\text{T}_{\text{E}}\text{X}$ as a platform on which *convenient work environments* could be built. One such work environment, \LaTeX , predominates today, and it is indeed convenient (kind of).

Nevertheless, \LaTeX is designed for all of us, so it is not surprising that we can improve on it for our personal use. There are many reasons to customize \LaTeX :

Goal 1 to enhance the readability of the source file;

Goal 2 to make notational and terminological changes easier;

Goal 3 to redefine names used by \LaTeX ;

Goal 4 to introduce consistent layouts.

There are many techniques to accomplish these.

Technique 1 Define commands and environments in order to enhance \LaTeX to meet your particular needs (see Sections 15.1, 15.1.7, and 15.2).

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-55281-6_15

Technique 2 Utilize delimited commands to write L^AT_EX documents in a more readable fashion (see Section 15.1.9).

Technique 3 Collect your frequently used commands into a command file (see Section 15.3).

Technique 4 Manipulate *counters*, integers—for instance, equation numbers and section numbers—and *length commands*, distance measurements—the `\voffset` command is an example (see Section 15.5).

Technique 5 Create customized list environments with the `list` environment (see Section 15.6).

Of course, there are many more reasons to customize and many more techniques to employ. We cover them in detail in this chapter.

We dedicate the last section to the pitfalls of customization (see Section 15.7). While the benefits of customization are great, there are many practices to avoid.

15.1 Custom commands

L^AT_EX provides hundreds of commands. Chances are good, however, that you still have specific needs that are not directly addressed by these commands. By judiciously adding *custom commands* (or *macros*) you can make your work more productive.

Custom commands follow the same rules as regular L^AT_EX commands (see Section 3.3.1).

15.1.1 Examples and rules

Commands to enhance readability

Let us start with a few examples of custom commands as shorthand for longer command(s) or text in order to enhance readability of the source file (Goal 1).

Example 1 If you use the `\leftarrow` command a lot, you could define

```
\newcommand{\larr}{\leftarrow}
```

Then you would only have to type `\larr` to obtain a left arrow.

Example 2 Instead of

```
\widetilde{a}
```

you could simply type `\wtilda` after defining

```
\newcommand{\wtilda}{\widetilde{a}}
```

I show you how to define a generalized version of such a command in Section 15.1.2.

Example 3 If you want to suppress the ligature in `iff` (see Section 3.4.6), you would normally have to type

```
if\textcompwordmark f
```

By defining a command `\Iff`,

```
\newcommand{\Iff}{if\textcompwordmark f}
```

you can type `\Iff` to get `iff`. We name this command `\Iff` because `\iff` is the symbol \iff (see Appendix A.4).

Example 4 If you use the construct $D^{[2]} \times D^{[3]}$ often, you could introduce the `\DxD` (D times D) command,

```
\newcommand{\DxD}{D^{[2]}\times D^{[3]}}
```

and then type `\DxD` instead of the longer, and hard to read, version throughout your document—serves also Goal 2.

Example 5 If you want to get a backslash in typewriter style, you would normally have to type (see Section 3.4.4)

```
\texttt{\symbol{92}}
```

Instead, you can introduce the `\bsl` command,

```
\newcommand{\bsl}{\texttt{\symbol{92}}}
```

and `\bsl` typesets as `\`.

Example 6 You can also use commands as a shorthand for text. For instance, if you use the phrase `subdirectly irreducible` many times in your document, you could define

```
\newcommand{\subdirr}{subdirectly irreducible}
```

`\subdirr` is now shorthand for `subdirectly irreducible`, which typesets as `subdirectly irreducible`.



Tip With modern editors, the need to have custom commands as shorthand is reduced. Most editors have “command completion” or “phrase completion”. For instance, in TeXShop, type the first few letters of a word and hit the escape key. The remaining letters are entered to match the first entry in the completion dictionary. Hitting escape again cycles through all possible completions.

To make this feature useful, you have to customize the completion dictionary. Note that command completion minimizes the number of characters you have to type, but it does not improve the readability of the source.

**Rule: Custom commands**

1. Issue the `\newcommand` command.
 2. In braces, type the name of your new command, for example, `\subdirr`, including the backslash (`\`).
 3. In a second pair of braces, define the command, in this example, `subdirectly irreducible`.
 4. Use the command as `\subdirr\` or `\subdirr{}` before a space, before an alphabetical character as `\subdirr{}`, and `\subdirr` otherwise.
-

Examples for Rule 4. For subdirectly irreducible lattice type

`\subdirr{}` lattice

or

`\subdirr\` lattice

and not `\subdirr lattice`. Indeed, typesetting `\subdirr lattice` results in `subdirectly irreduciblelattice`. By the first spacing rule, `\subdirr\lattice` is not any better (see Section 3.2.1). If you want subdirectly irreducibles, you must use the `\subdirr{}` form. Indeed, `\subdirr{}`s typesets as subdirectly irreducibles.

Using new commands

It is good practice to place custom commands in the preamble of your document or in a command (style) file you load with a `\usepackage` command (see Section 15.3)—provided that you do not submit to a journal that does not allow this. Then you always know where to look for the command definitions. An exception is a custom command that you want to restrict to a part of the document. Delimit the segment with braces and define the custom command within those braces (see Section 3.3.2). Instead of a pair of braces, you can use `\begingroup` and `\endgroup`, which is easier to see. Section 15.2.5 recommends yet another approach.

**Tip**

- If errors occur, isolate the problem. Comment out the custom commands and reintroduce them one at a time.
 - \LaTeX only checks whether the braces match in the command definition. Other mistakes are found only when the command is used.
-

For instance, if you define a command with a spelling error

```
\newcommand{\bfA}{\textf{A}}
```

then at the first use of `\bfA` you get the message

```
! Undefined control sequence.
\bfA ->\textf
      {A}
```

Note that L^AT_EX is not complaining about `\bfA` but about the misspelled `\textbf` command in the definition of `\bfA`.

Be careful not to define a custom command with a name that is already in use. If you do, you get a message such as

```
! LaTeX Error: Command \larr already defined.
```

To correct the error, replace the command name with a new one. On the other hand, if you need to replace an existing command, you have to *redefine* it. See Section 15.1.5 for how to do so.



Tip Use spaces to make your source files more readable, but avoid them in definitions.

For example, you may type

```
$D^{\ \langle 2 \ \rangle } + 2 = x^{\ \mathbf{a} \ }$
```

This may help you see how the braces match, easily identify relations and operations, and so on. *Do not add these spaces in command definitions* because it may result in unwanted spaces in your typeset document. You may start a new line to increase the readability of a command definition, provided that you terminate the previous line with `%`. For instance, borrowing an example from page 412:

```
\newcommand{\Xquotphi}[2]{%
  \dfrac{\varphi \cdot X_{n, #1}}{
    {\varphi_{#2}\times \varepsilon_{#1}}}
```



Tip In the definition of a new command, command declarations need an extra pair of braces (see Section 3.3.3).

Say you want to define a command that typesets the warning: *Do not redefine this variable!* It is very easy to make the following mistake:

```
\newcommand{\Warn}{\em Do not redefine this variable!}
```

`\Warn` typesets the warning emphasized, but everything that follows the warning is also emphasized (more precisely, until the end of the `\Warn` command's scope). Indeed, `\Warn` is replaced by `\em Do not redefine this variable!` so the effect of `\em` goes beyond the sentence to the next closing brace.

The correct definition is

```
\newcommand{\Warn}{\em Do not redefine this variable!}}
```

Even simpler, you could use a command with an argument

```
\newcommand{\Warn}{\emph{Do not redefine this variable!}}
```



Tip There are limits as to what can be done with custom commands. For instance, you cannot introduce `\bal` for `\begin{align}` and `\eal` for `\end{align}`. So if you want to introduce a new type of custom command, do just one and try it out.

The xspace package

Rule 4 (on page 406) is the source of many annoying problems in \LaTeX . David Carlisle's `xspace` package helps eliminate such problems. In the preamble, load the package with

```
\usepackage{xspace}
```

Whenever you define a command that may have such problems, add the `\xspace` command to the definition. For instance, define `\subdirr` as

```
\newcommand{\subdirr}{subdirectly irreducible\xspace}
```

Then all the following typesets `subdirectly irreducible lattice` correctly:

```
\subdirr\_\lattice
\subdirr{\_\lattice
\subdirr\_\lattice
```

Note that `\xspace` does not add space if followed by a punctuation mark, so to get

```
┌
└ the lattice is subdirectly irreducible.
└
type
the lattice is \subdirr.
```



Tip Be careful not to use `\xspace` twice in a definition.

For instance, if you define

```
\newcommand{\tex}{\TeX\xspace}
\newcommand{\bibtex}{\textsc{Bib}\kern-.1em\tex\xspace}% Bad!!!
then
\begin{verbatim}
\bibtex, followed by a comma
```

typesets as

```
┌
BIBTEX , followed by a comma
└
```

The correct definitions are

```
\newcommand{\tex}{\TeX\xspace}
\newcommand{\bibtex}{\textsc{Bib}\kern-.1em\TeX\xspace}% Correct!
```

Of course, if you want to get \TeX book, you cannot use the `\xspace` variant definition of `\tex`.

Ensuring math

The `\ensuremath` command is useful for defining commands for both text and math mode. Suppose you want to define a command for $D^{(2)}$. If you define it as

```
\newcommand{\Dsqr}{D^{\langle2\rangle}}
```

then you can use the command in math mode, but not in text mode. If you define it as

```
\newcommand{\Dsqr}{\$D^{\langle2\rangle}\$}
```

then it works in text mode, but not in math mode. Instead, define this command as

```
\newcommand{\Dsqr}{\ensuremath{D^{\langle2\rangle}}}
```

Then `\Dsqr` works correctly in both contexts.

This example also shows the editorial advantages of custom commands. Suppose the referee suggests that you change the notation to $D^{[2]}$. To carry out the change you only have to change one line:

```
\newcommand{\Dsqr}{\ensuremath{D^{[2]}}}
```

It is hard to overemphasize the importance of this example. You may want to change notation because:

- you found a better notation;
- your coauthor insists;
- your article appears in a conference proceedings, and the editor wants to unify the notation;
- you are reusing the code from this article in another one or in a book, where the notation is different.

15.1.2 Arguments

Arguments of custom commands work the same way as for \LaTeX commands (see page 55). Define

```
\newcommand{\fsqAB}{(f^2)^{[[\frac{A^2}{B-1}]]}}
```

Then `\fsqAB` typesets as $(f^2)^{[\frac{A^2}{B-1}]}$ in a math formula. If you want to use `\fsqAB` in math and also by itself in text, define it with `\ensuremath`, as

```
\newcommand{\fsqAB}{\ensuremath{(f^2)^{[[\frac{A^2}{B-1}]]}}}
```

However, if you use this construct for many functions f , then you may need a generalized command, such as

```
\newcommand{\sqAB}[1]{\ensuremath{(#1^2)^{[[\frac{A^2}{B-1}]]}}}
```

Now `\sqAB{g}` typesets $(g^2)^{[\frac{A^2}{B-1}]}$. The form of this `\newcommand` is the same as before, except that after the name of the command in braces, `\sqAB`, we specify the number of arguments in brackets (in this example, `[1]`). Then we can use `#1` in the definition of the command. When the command is invoked, the argument you provide replaces `#1` in the definition. Typing `\sqAB{q}` results in the formula $(q^2)^{[\frac{A^2}{B-1}]}$, while `\sqAB{r}` gives $(r^2)^{[\frac{A^2}{B-1}]}$.

Notice how these examples disrupt the normal spacing between lines—a practice to avoid!

A custom command may have up to nine arguments, numbered 1–9.

The simplest examples just allow you to invoke an existing command under a new name. For instance, the `\eqref` command introduced in Section 7.3 to reference equations (the equation number upright, enclosed in parentheses), would also be useful to reference items for the custom list environment `enumeratei` introduced in Section 15.2.1—see Example 3. Indeed, for the `enumeratei` environment, we want

references to items to be typeset upright, enclosed in parentheses. So if the first item has label `First`, we could reference it with `\eqref{First}`, which typesets as (i). But `\eqref{First}` seems awkward and inappropriate; we are referencing an item not an equation.

So define

```
\newcommand{\itemref}[1]{\eqref{#1}}
```

and now we can reference the first item with `\itemref{First}`, which typesets as (i).

Following are some simple examples of custom commands with arguments.

Example 1 In the preamble of the source file for this book, I defined

```
\newcommand{\env}[1]{\textnormal{\texttt{#1}}}
```

In this example, the `\env` command is used to typeset environment names. So the environment name `center` is typed as

```
\env{center}
```

Again the editorial advantage is obvious. If the editor wants the environment names set in sans serif, only one line in the book has to be changed to alter every occurrence of a typeset environment name:

```
\newcommand{\env}[1]{\textsf{#1}}
```

Example 2 An argument (e.g., `#1`) may occur more than once in a definition. A natural example is provided by the `\index` command (see Section 18.1). Typically, if you wanted to include a phrase, say `subdirectly irreducible lattice`, in your index, you would have to type

```
this proves that  $L$  is a subdirectly irreducible lattice%
\index{subdirectly irreducible lattice}
```

You could instead define an “index it” command such as

```
\newcommand{\indexit}[1]{#1\index{#1}}
```

The argument of this command is a phrase to be both typeset and included in the index. Using this command, you can type

```
it is a \indexit{subdirectly irreducible lattice}
```

If you want all such index entries to be typeset in italics, then `\indexit` should be defined as

```
\newcommand{\indexit}[1]{#1\index{#1@\textit{#1}}}
```

in which `#1` occurs three times. (See Chapter 18 for more information about index commands.)

Example 3 Let us define a command with three arguments for congruences:

```
\newcommand{\congr}[3]{#1\equiv#2\pod{#3}}
```

Now type `\congr{a}{b}{\theta}` to typeset $a \equiv b \ (\theta)$. In Section 15.1.9, I present another command for typesetting congruences.

Example 4 In the `art2.tex` article (see Section 11.3), there are a lot of vectors with only one nonzero entry:

$$\langle \dots, 0, \dots, \overset{i}{d}, \dots, 0, \dots \rangle$$

the i above the d indicates that it is the i th component of the vector. A command `\vectsup`, a vector with a superscript, producing this symbol can be defined as

```
\newcommand{\vectsup}[2]{\langle \dots, 0, \dots,
\overset{#1}{#2}, \dots, 0, \dots\rangle}
```

`\vectsup{i}{d}` in a math formula now produces $\langle \dots, 0, \dots, \overset{i}{d}, \dots, 0, \dots \rangle$.

Formula 20 of the *Formula Gallery* (Section 7.10),

$$\mathbf{A} = \begin{pmatrix} \frac{\varphi \cdot X_{n,1}}{\varphi_1 \times \varepsilon_1} & (x + \varepsilon_2)^2 & \cdots & (x + \varepsilon_{n-1})^{n-1} & (x + \varepsilon_n)^n \\ \frac{\varphi \cdot X_{n,1}}{\varphi_2 \times \varepsilon_1} & \frac{\varphi \cdot X_{n,2}}{\varphi_2 \times \varepsilon_2} & \cdots & (x + \varepsilon_{n-1})^{n-1} & (x + \varepsilon_n)^n \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\varphi \cdot X_{n,1}}{\varphi_n \times \varepsilon_1} & \frac{\varphi \cdot X_{n,2}}{\varphi_n \times \varepsilon_2} & \cdots & \frac{\varphi \cdot X_{n,n-1}}{\varphi_n \times \varepsilon_{n-1}} & \frac{\varphi \cdot X_{n,n}}{\varphi_n \times \varepsilon_n} \end{pmatrix} + \mathbf{I}_n$$

is a good candidate for custom commands. By defining

```
\newcommand{\Xquotphi}[2]{%
\dfrac{\varphi \cdot X_{n, #1}}{\varphi_{#2} \times \varepsilon_{#1}}}
\newcommand{\exn}[1]{(x + \varepsilon_{#1})^{#1}}
```

the two new commands,

```
\[
\Xquotphi{2}{3} \quad \exn{n}
\]
```

are typeset as

$$\frac{\varphi \cdot X_{n,2}}{\varphi_3 \times \varepsilon_2} \quad (x + \varepsilon_n)^n$$

With these custom commands, you can rewrite Formula 20 as follows:

```
\[
\mathbf{A} =
\begin{pmatrix}
```

```

\Xquotphi{1}{1} & \exn{2} & \cdots & \exn{n - 1}+
& \exn{n}\\[10pt]
\Xquotphi{1}{2} & \Xquotphi{2}{2} & \cdots
& \exn{n - 1} & \exn{n}\\
\hdotsfor{5}\\
\Xquotphi{1}{n} & \Xquotphi{2}{n} & \cdots &
\Xquotphi{n - 1}{n} & \Xquotphi{n}{n}
\end{pmatrix}
+ \mathbf{I}_{\{n\}}
\]
```

Observe how much shorter this form is than the version shown in the *Formula Gallery* and how much easier it is to read. It is also easier to reuse in a subsequent article.

15.1.3 Short arguments

There are three ways of defining new commands:

```
\newcommand \renewcommand \providecommand
```

We take up the last two in Section 15.1.5. They define commands that can take any number of paragraphs as arguments. The *-ed versions of these commands define *short* commands (see Section 3.3.3) that take a block of text that contains no paragraph break as an argument. For instance,

```
\newcommand{\bigbold}[1]{\large\bfseries#1}
```

makes its argument large and bold. So

```
\bigbold{First paragraph.
```

```
Second paragraph.
```

```
}
```

```
prints
```

```
┌
```

First paragraph.

Second paragraph.

```
└
```

as expected. On the other hand, if you define

```
\newcommand*\bigbold[1]{\large\bfseries#1}
```

and then attempt to typeset the previous example, you get the message

Paragraph ended before `\bigbold` was complete.
`<to be read again>`

```
\par
\par
\par
```

Short commands are often preferable because of their improved error checking.

15.1.4 Optional arguments

We can define a command whose first argument is *optional*, and we provide a *default value* for this optional argument. To illustrate, let us define the command

```
\newcommand{\SimpleSum}{a_{1}+a_{2}+\dots+a_{n}}
```

`\SimpleSum` now produces $a_1 + a_2 + \dots + a_n$. Now we change this command so that we can sum from 1 to m if necessary, with n as the default:

```
\newcommand{\BetterSum}[1][n]{a_{1}+a_{2}+\dots+a_{\#1}}
```

`\BetterSum` still produces $a_1 + a_2 + \dots + a_n$, but `\BetterSum[m]` typesets as $a_1 + a_2 + \dots + a_m$.

A `\newcommand` may have up to nine arguments, but *only the first* may be optional. The following command has two arguments, one optional:

```
\newcommand{\BestSum}[2][n]{\#2_{1}+\#2_{2}+\dots+\#2_{\#1}}
```

Now

<code>\BestSum{a}</code>	typesets as	$a_1 + a_2 + \dots + a_n$
<code>\BestSum{b}</code>	typesets as	$b_1 + b_2 + \dots + b_n$
<code>\BestSum[m]{c}</code>	typesets as	$c_1 + c_2 + \dots + c_m$

15.1.5 Redefining commands

L^AT_EX makes sure that you do not inadvertently define a new command with the same name as an existing command (see, for example, page 407). Assuming that you have already defined the `\larr` command as in Section 15.1.1 (to typeset \leftarrow), to *redefine* `\larr`, use `\renewcommand`:

```
\renewcommand{\larr}{\Longleftarrow}
```

and now `\larr` typesets as \Longleftarrow .



Tip Use the `\renewcommand` command sparingly and make sure that you understand the consequences of redefining an existing command. Redefining L^AT_EX commands may cause L^AT_EX to behave in unexpected ways, or even crash.

Blind redefinition is the route to madness.

See also the discussion in Section 15.7.

You can also use `\renewcommand` to redefine commands defined by L^AT_EX or any package. For instance, the end of proof symbol, `\qedsymbol`, used by the `proof` environment, can be changed to the solid black square some people prefer (defined in the `amssymb` package) with the command

```
\renewcommand{\qedsymbol}{$\blacksquare$}
```

Even better, define

```
\renewcommand{\qedsymbol}{\ensuremath{\blacksquare}}
```

so that you can use `\qedsymbol` in both text and math mode. Section 15.1.7 has more on redefining names.

`\renewcommand` has a companion, `\providecommand`. If the command it defines has already been defined, the original command is left unchanged. Otherwise, the `\providecommand` command acts exactly like `\newcommand`. For instance, the `\bysame` command (see Section 10.5.1, page 274) is defined in some document classes as

```
\newcommand{\bysame}{\makebox[3em]{\hrulefill}\thinspace}
```

If you want to use the `\bysame` command in your bibliography and include this definition in your document, L^AT_EX generates a message when you typeset your document using a document class that already defines `\bysame` (all AMS document classes do). However, if you define `\bysame` in your document using `\providecommand`:

```
\providecommand{\bysame}{%
  \makebox[3em]{\hrulefill}\thinspace}
```

the `\bysame` command typesets correctly whether or not the document class defines it.

15.1.6 Defining operators

The powerful `\DeclareMathOperator` command defines a new operator:

```
\DeclareMathOperator{\opCommand}{opName}
```

Invoke the new operator with `\opCommand`, which is then typeset with `opName`.



Rule: The `\DeclareMathOperator` command must be placed in the preamble.

For example, to define the operator `Truncat`, invoked by the command `\Trunc`, place this in the preamble:

```
\DeclareMathOperator{\Trunc}{Truncat}
```

An operator is typeset in math roman with a little space after it, so `$_\Trunc A$` typesets as `Truncat A`.

The second argument is typeset in math mode but `-` and `*` are typeset as they would be in text. Here are some more examples. Define in the preamble two operators:

```
\DeclareMathOperator{\Trone}{Trunc_{1}}
\DeclareMathOperator{\Ststar}{Star-one*}
```

Then in the body of the article

`$_\Trone A$` is typeset as `Truncat1 A`

`$_\Ststar A$` is typeset as `Star-one* A`

To define an operator with limits, use the `*-ed` form

```
\DeclareMathOperator*{\doublesum}{\sum\sum}
```

and then (see Section 7.6.4 for multiline subscripts)

```
\[
  \doublesum_{\begin{subarray}{l}
    i^2+j^2 = 50\\
    i,\ j \leq 10
  \end{subarray}}
  \frac{x^i + y^j}{(i + j)!}
\]
```

typesets as

$$\sum_{\substack{i^2+j^2=50 \\ i, j \leq 10}} \sum \frac{x^i + y^j}{(i + j)!}$$

15.1.7 Redefining names

A number of names, such as `Table`, `List of Tables`, `Abstract`, and so on, are typeset in your document by \LaTeX . You can easily change these names.

For instance, if you are preparing your manuscript for the proceedings of a meeting, and `Abstract` has to be changed to `Summary`, you can do so with

```
\renewcommand{\abstractname}{Summary}
```

Table 15.1 lists the commands that define such names in various document classes, along with their default definitions and the major document classes using the commands. It is easy to check whether your document class defines such a command, simply open the appropriate `cls` file and search for the command.

If your document has photographs rather than figures, you could redefine

```
\renewcommand{\figurename}{Photograph}
\renewcommand{\listfigurename}{List of Photographs}
```

Command	Default Value	Defined by Document Class
<code>\abstractname</code>	Abstract	aa, ab, ap, a, p, r
<code>\appendixname</code>	Appendix	aa, ab, ap, a, b, r
<code>\bibname</code>	Bibliography	ab, b, r
<code>\ccname</code>	Cc	l
<code>\chaptername</code>	Chapter	ab, b, r
<code>\contentsname</code>	Contents	aa, ab, ap, a, b, r
<code>\datename</code>	Date	aa, ab, ap
<code>\enclname</code>	Enclosure	l
<code>\figurename</code>	Figure	aa, ab, ap, a, b, r
<code>\headtoname</code>	To	l
<code>\indexname</code>	Index	aa, ab, ap, a, b, r
<code>\keywordsname</code>	Key words and phrases	aa, ab, ap
<code>\listfigurename</code>	List of Figures	aa, ab, ap, a, b, r
<code>\listtablename</code>	List of Tables	aa, ab, ap, a, b, r
<code>\pagename</code>	Page	l, p
<code>\partname</code>	Part	aa, ab, ap, a, b, r
<code>\proofname</code>	Proof	aa, ab, ap
<code>\refname</code>	References	aa, ap, a
<code>\see</code>	see	aa, ab, ap
<code>\seealso</code>	see also	aa, ab, ap
<code>\subjclassname</code>	1991 Mathematics Subject Classification	aa, ab, ap
<code>\subjclassname[2010]</code>	2010 Mathematics Subject Classification	aa, ab, ap
<code>\tablename</code>	Table	aa, ab, ap, a, b, r

Document class codes: aa `amsart`, ab `amsbook`, ap `amsproc`,
a `article`, b `book`, l `letter`, p `proc`, and r `report`

Table 15.1: Redefinable name commands in \LaTeX .

15.1.8 Showing the definitions of commands

If you are defining a new command with `\newcommand` and an error message informs you that the command name is already in use, then you may want to find out the existing definition of the command. For instance, if you call a new command `\vec`, you would get the message

```
! LaTeX Error: Command \vec already defined.
```

You can find out the definition of the `\vec` command by getting into interactive mode and typing

```
*\show \vec
```

LaTeX responds with

```
> \vec=macro:
->\mathaccent "017E .
<*> \show \vec
```

informing you that `\vec` is a command, and, specifically, a math accent (see Sections 7.7 and A.7). Now try `\hangafter` (see Section 3.6.2):

```
*\show \hangafter
```

```
> \hangafter=\hangafter.
```

The response indicates that `\hangafter` is a *primitive*, defined by TeX itself. Redefining a primitive is not a good idea.

Try one more command, `\medskip` (see Section 3.7.2), to find out how large it is:

```
*\show \medskip
> \medskip=macro:
->\vspace \medskipamount .
```

The third line indicates that the length is stored in `\medskipamount`. If we use `\show` to ask what `\medskipamount` is defined to be:

```
*\show \medskipamount
> \medskipamount=\skip14.
```

we do not get a very useful answer. `\medskipamount` is unlike most of the commands you have seen so far. It is a *length command* (see Section 15.5.2), containing the value of `\medskip`. You can ask for the value of a length command (or parameter) with the `\showthe` command:

```
*\showthe \medskipamount
```

```
> 6.0pt plus 2.0pt minus 2.0pt.
```

So `\medskip` is a vertical space of 6 points that can stretch or shrink by up to 2 points.

L^AT_EX has many registers that contain numbers:

- counters containing integers, such as 3;
- dimensions such as 10.2pt, for example, `\textwidth` (see Section 10.6);
- lengths, written in the form 6.0pt plus 2.0pt minus 2.0pt, also called a *glue* or a *rubber length* (see Section 15.5.2).

Use the `\showthe` command to display the value for any of these registers.

You can also type the `\show` and `\showthe` commands directly into your document rather than go into interactive mode. L^AT_EX's response appears in the log window, and is saved into the log file.

Of course, we should point out in this edition of this book that you can alternatively use ChatGPT to find out the definition of a L^AT_EX command. Ask ChatGPT:

latex `\vec`

and it responds:

You can use the `\vec` command to typeset a vector with an arrow above it to indicate that it is a vector.

The LaTeX command for creating a vector with an arrow above the symbol is `\vec`. You place the symbol or letter representing the vector inside the curly braces. For example, to represent a vector "a", you would write `\vec{a}`.

15.1.9 Delimited commands

You can define new commands in T_EX using characters and symbols to delimit arguments. Such *delimited commands* provide a way to write more readable source documents.

First we have to learn how to define a command using T_EX's `\def` command. Type `\def`, followed by the new command name (not in braces), then the definition in braces. For example, the first command defined in Section 15.1.1,

```
\newcommand{\larr}{\leftarrow}
```

could be typed

```
\def\larr{\leftarrow}
```

T_EX's `\def` command does not check whether a new command name is already in use, so `\def` behaves differently from the L^AT_EX's `\newcommand`, `\renewcommand`, and `\providecommand` (see Section 15.1.5). If the `\larr` command was defined previously, the original definition is overwritten.



Tip It is your responsibility to ensure that your command name is unique when you define

a command using `\def`. \LaTeX provides no protection. Use the techniques introduced in Section 15.1.8 to check a name before you define a command with `\def`.

Now we can start discussing delimited commands with a simple example, defining a command for vectors:

```
\def\vect<#1>{\langle#1\rangle}
```

Note that `\vect` is a command with one argument, `#1`. When invoked, it typesets \langle , the argument, and then \rangle .

In the definition of `\vect`, the argument `#1` is delimited by `<` and `>`. When the command is invoked, the argument must be delimited the same way. So to typeset the vector $\langle a, b \rangle$, we invoke `\vect` with

```
\vect<a,b>
```

This looks somewhat like a vector, and the name `\vect` serves as a reminder.

You have to be careful with delimited commands because the math spacing rules (see Section 7.2) do not hold in either the definition or the invocation. So if there is a space before `#1`, in the definition of `\vect`,

```
\def\vect< #1>{\langle#1\rangle}
```

then `$_\vect<a,b>$` results in the message

```
! Use of \vect doesn't match its definition.
1.12 $_\vect<a
      ,b>$
```

which is clear enough. If the space is on the other side of the `#1`, as in

```
\def\vect<#1 >{\langle#1\rangle}
```

the message is slightly more confusing:

```
Runaway argument?
a,b>$
! Paragraph ended before \vect was complete.
<to be read again>
      \par
```

The moral is that if you use delimited commands, you must be very careful that each invocation exactly matches the definition. In Example 3 of Section 15.1.2, we introduced a command with three arguments for typing congruences:

```
\newcommand{\congr}[3]{#1\equiv#2\pod{#3}}
```

`\congr{a}{b}{\theta}` produces $a \equiv b (\theta)$. This command is easy to remember, but it does not make the source file easily readable. For that, we use a delimited command. Let us redo the congruence example with a delimited command

```
\def\congr#1=#2(#3){#1\equiv#2\pod{#3}}
```

so that `\congr a=b(\theta)` produces $a \equiv b (\theta)$. In the source document, the formula `\congr a=b(\theta)` looks a bit like the typeset congruence and it is easier to read. I included this definition in the `newlattice.sty` command file (see Section 15.3).

There is only one catch. Suppose you want to typeset the formula

$$x = a \equiv b (\theta)$$

If you type `\congr x=a=b(\theta)`, \LaTeX typesets it as $x \equiv a = b (\theta)$. Indeed, `x` is delimited on the right by the first `=`, so \LaTeX believes that the first argument is `x`. The second argument is delimited by the first `=` and the left parenthesis, so it is `a=b`. In such cases, you can help \LaTeX find the correct first argument by enclosing it in braces:

```
\congr{x=a}=b(\theta)
```

Here is our final example. In Section 3.3.1 we discuss the problem of typing a command such as `\TeX` (the example there was `\today`) in the form `\TeX__` so that it is typeset as a separate word. The problem is that if you type `\TeX` without the trailing `__`, `\TeX` is merged with the next word, and there is no message to warn you. One solution is to use a delimited command:

```
\def\tex/{\TeX}
```

Now to get \TeX , type `\tex/`. If a space is needed after it, type `\tex/__`. If you forget the closing `/`, you get a message.

A better solution to this problem is the use of the `xspace` package—provided you do not want to typeset something like \TeX book (see Section 15.1.1). However, many documents use the delimited construct (including the AMS documentation), so you should be familiar with it.

15.2 Custom environments

Most custom commands are new commands. *Custom environments*, as a rule, are built on existing environments. We start with such custom environments (Section 15.2.1) and then proceed to investigate:

- arguments (Section 15.2.2),
- optional arguments (Section 15.2.3),
- short arguments (Section 15.2.4).

Finally, we discuss how to define brand-new environments (Section 15.2.5).

15.2.1 Modifying existing environments

If you do not like the name of the `proof` environment and would prefer to use the name `demo`, define

```
\newenvironment{demo}
  {\begin{proof}}
  {\end{proof}}
```

Note that this does not change how the environment is typeset, only the way it is invoked.

To modify an existing environment, `oldenv`, type

```
\newenvironment{name}
  {begin_text}
  {end_text}
```

where *begin_text* contains the command `\begin{oldenv}` and *end_text* contains the command `\end{oldenv}`.



Tip Remember not give a new environment the name of an existing command or environment.

For instance, if you define

```
\newenvironment{parbox}
  {...}
  {...}
```

you get the message

```
! LaTeX Error: Command \parbox already defined.
```

If there is an error in such a custom environment, the message generated refers to the environment that was modified, not to your environment. For instance, if you misspell `proof` as `prof` when you define

```
\newenvironment{demo}
  {\begin{prof}}
  {\end{proof}}
```

then *at the first use* of the `demo` environment you get the message

```
! LaTeX Error: Environment prof undefined.
```

```
1.13 \begin{demo}
```

If you define

```
\newenvironment{demo}
  {\begin{proof}\em}
  {\end{proof}}
```

at the first use of `demo` you get the message

```
! LaTeX Error: \begin{proof} on input line 5
    ended by \end{proof}.
1.14 \end{demo}
```

Here are four more examples of modified environments.

Example 1 The command

```
\newenvironment{demo}
  {\begin{proof}\em}
  {\end{proof}}+
```

defines a `demo` environment that typesets an emphasized proof. Note that the scope of `\em` is the `demo` environment.

Example 2 The following example defines a very useful environment. It takes an argument to be typeset as the name of a theorem:

```
\newtheorem*{namedtheorem}{\theoremname}
\newcommand{\theoremname}{testing}
\newenvironment{named}[1]{
  \renewcommand{\theoremname}{#1}
  \begin{namedtheorem}}
  {\end{namedtheorem}}
```

For example,

```
\begin{named}{Name of the theorem}
Body of theorem.
\end{named}
```

produces

Name of the theorem. *Body of theorem.*

in the style appropriate for the `\newtheorem*` declaration. This type of environment is often used to produce an unnumbered **Main Theorem** (see Section 15.4) or when typesetting an article or book in which the theorem numbering is already fixed, for instance, when publishing a book in \LaTeX that was originally typeset by another typesetting system.

Example 3 In Section 4.2.4, we came across the `enumerate` package, which allows you to customize the `enumerate` environment. If the `enumerate` package is loaded, you can invoke the `enumerate` environment with an optional argument specifying how the counter should be typeset, for instance, with the option `[\upshape (i)]`,

```
\begin{enumerate}[\upshape (i)]
  \item First item\label{First}
\end{enumerate}
```

items are numbered (i), (ii), and so on. So now we define

```
\newenvironment{enumeratei}{\begin{enumerate}%
  [\upshape (i)]}%
  {\end{enumerate}}
```

and we can invoke the new environment with (see Sections 15.3 and 15.4)

```
\begin{enumeratei}
  \item \label{ }
\end{enumeratei}
```

Reference items in the `enumeratei` environment with the `\itemref` command introduced in Section 15.1.2.

Example 4 If you want to define an environment for displaying text that is numbered as an equation, you might try

```
\newenvironment{texreqn}
  {\begin{equation} \begin{minipage}{0.9\linewidth}}
  {\end{minipage} \end{equation}}
```

But there is a problem. If you use this environment in the middle of a paragraph, an interword space appears at the beginning of the first line after the environment. To remove this unwanted space, use the `\ignorespacesafterend` command, as in

```
\newenvironment{texreqn}
  {\begin{equation} \begin{minipage}{0.9\linewidth}}
  {\end{minipage} \end{equation} \ignorespacesafterend}
```

Examples 2 and 3 are included in the `newlattice.sty` command file (see Section 15.3).

See Section 15.6.3 for custom lists as custom environments.

Redefine an existing environment with the `\renewenvironment` command. It is similar to the `\renewcommand` command (see Section 15.1.5).



Tip There are some environments you cannot redefine; for instance, the `verbatim` environment and all the AMS multiline math environments.

15.2.2 Arguments

An environment defined by the `\newenvironment` command can take arguments (see Example 2 in Section 15.2.1), but they can only be used in the *begin_text* argument of the `\newenvironment` command. Here is a simple example. Define a theorem proclamation in the preamble (see Section 4.4), and then define a theorem that can be referenced:

```
\newenvironment{theoremRef}[1]
  {\begin{theorem}\label{T:#1}}
  {\end{theorem}}
```

This is invoked with

```
\begin{theoremRef}{label}
```

The `theoremRef` environment is a modified environment. It is a theorem that can be referenced (with the `\ref` and `\pageref` commands, of course) and it invokes the `theorem` environment when it defines `T:label` to be the label for cross-referencing.

15.2.3 Optional arguments with default values

The first argument of an environment created with the `\newenvironment` command may be an *optional argument with a default value*. For example,

```
\newenvironment{narrow}[1][3in]
  {\noindent\begin{minipage}{#1}}
  {\end{minipage}}
```

creates a narrow environment. By default, it sets the body of the environment in a 3-inch wide box, with no indentation. So

```
\begin{narrow}
```

```
This text was typeset in a \texttt{narrow}
  environment, in a 3-inch wide box, with no indentation.
\end{narrow}
```

typesets as

```
┌ This text was typeset in a narrow environment, in
  a 3-inch wide box, with no indentation.
└
```

You can also give an optional argument to specify the width. For example,

```
\begin{narrow}[3.5in]
  This text was typeset in a \texttt{narrow} environment,
  in a 3-inch wide box, with no indentation.
\end{narrow}
```

which produces the following false statement:

```
┌ This text was typeset in a narrow environment, in a 3-inch
└ wide box, with no indentation.
```

15.2.4 Short contents

We have discussed two commands that define new environments,

```
\newenvironment and \renewenvironment
```

These commands allow you to define environments whose contents (*begin_text* and *end_text*; see page 422) can include any number of paragraphs. The *-ed versions of these commands define *short* environments whose contents cannot contain a paragraph break (a blank line or a `\par` command).

15.2.5 Brand-new environments

Some custom environments are not modifications of existing environments:

Example 1 A command remains effective only within its scope (see Section 3.3.2). Now suppose that you want to make a change, say redefining a counter, for only a few paragraphs. You could simply place braces around these paragraphs, but they are hard to see. So define

```
\newenvironment{exception}
  {\relax}
  {\relax}
```

and then

```
\begin{exception}
  new commands
  body
\end{exception}
```

The environment stands out better than a pair of braces, reminding you later about the special circumstances. The `\relax` command does nothing, but it is customary to include a `\relax` command in such a definition to make it more readable.

Example 2 In this example, we define a new environment that centers its body vertically on a new page:

```
\newenvironment{vcenterpage}
  {\newpage\vspace*{\fill}}
  {\vspace*{\fill}\par\pagebreak}
```

For `\vspace`, see Section 3.7.2 and for `\fill`, see the last subsection in Section 15.5.

15.3 A custom command file

Custom commands, of course, are a matter of individual need and taste. I have collected some commands for writing papers in lattice theory in the `newlattice.sty` file, which you can find in the `samples` folder (see page 5). I hope that this model helps you to develop a command file of your own. Please remember that everything we discuss in this section is a reflection of *my* work habits. Many experts disagree with one or another aspect of the way I define the commands, so take whatever suits your needs. And keep in mind the dangers of customization discussed in Section 15.7.



Tip Some journals do not permit the submission of a separate custom command file. For such journals, just copy the needed custom commands into the preamble of the article.

This file is named `newlattice.sty`. It can be loaded with `\usepackage`. This has a number of advantages.

Your command names should be mnemonic. If you cannot easily remember a command's name, rename it. The implication here is that your command file should not be very large unless you have an unusual ability to recall abbreviations.

Here are the first few lines of the `newlattice.sty` command file:

```
%newlattice.sty
% Custom command file for lattice papers
\ProvidesPackage{newlattice}[2022/03/15 v1.6]
\RequirePackage{fixltx2e}
\RequirePackage{amsmath}
\RequirePackage{amssymb}
\RequirePackage{latexsym}
\RequirePackage{eucal}
\RequirePackage{verbatim}
\RequirePackage{enumerate}
\RequirePackage{xspace}
```

The line

```
\ProvidesPackage{newlattice}[2022/03/15 v1.6]
```

gives a message when a document loads `newlattice`; it is written in the log file.

The next seven lines name the required packages. It loads any of these packages that have not yet been loaded. A package already loaded is not loaded in again.

Being able to specify the packages we need is one of the great advantages of command files. When I write a document, the packages are already there if I need them.

You may want some justification for the inclusion of two of these packages in this list. The `verbatim` package is on the list so that I can use the `comment` environment to comment out large blocks of text (see Section 3.5.1), which is useful for finding errors and typesetting only parts of a longer document.



Tip Do not forget to remove your comments before submission.

The `enumerate` package is on the list because the `enumeratei` and `enumeratea` environments, defined in `newlattice.sty`, require it.

If you start your article with

```
\documentclass{amsart}
\usepackage{newlattice}
```

then the `\listfiles` command in the preamble produces the following list in the log file when your document is typeset:¹

```
*File List*
amsart.cls      2020/05/29 v2.20.6
amsmath.sty     2022/04/08 v2.17n AMS math features
amstext.sty     2021/08/26 v2.01 AMS text
amsgen.sty     1999/11/30 v2.0 generic functions
amsbsy.sty     1999/11/29 v1.2d Bold Symbols
amsopn.sty     2022/04/08 v2.04 operator names
umsa.fd        2013/01/14 v3.01 AMS symbols A
amsfonts.sty   2013/01/14 v3.01 Basic AMSFonts support
newlattice.sty 2011/07/14 Commands for lattices based on LTF
amssymb.sty    2013/01/14 v3.01 AMS font symbols
latexsym.sty   1998/08/17 v2.2e Standard LaTeX package
                                     (lasy symbols)
eucal.sty      2009/06/22 v3.00 Euler Script fonts
verbatim.sty   2022-07-02 v1.5u LaTeX2e package
               †for verbatim enhancements
enumerate.sty  2015/07/23 v3.00 enumerate extensions (DPC)
xspace.sty     2014/10/28 v1.13 Space after command names
graphicx.sty   2021/09/16 v1.2d Enhanced LaTeX Graphics
keyval.sty     2022/05/29 v1.15 key=value parser (DPC)
trig.sty       2021/08/11 v1.11 sin cos tan (DPC)
```

¹The list for this book is more than 10 pages long...

```

graphics.cfg      2016/06/04 v1.11 sample graphics configuration
pdftex.def       2022/09/22 v1.2b Graphics/color driver for pdftex
l3backend-pdftex.def 2023-01-16 L3 backend support:
                        PDF output (pdfTeX)
    umsa.fd       2013/01/14 v3.01 AMS symbols A
    umsb.fd       2013/01/14 v3.01 AMS symbols B
    ulasy.fd      1998/08/17 v2.2e LaTeX symbol font definitions
supp-pdf.mkii
epstopdf-base.sty 2020-01-24 v2.11 Base part for package epstopdf
epstopdf-sys.cfg   2010/07/13 v1.3 Configuration of epstopdf
                        for TeX Live

```

This provides a list of all packages already invoked.

Now we continue with `newlattice.sty`. After the introductory section dealing with \LaTeX and the packages, we define some commands for writing about lattices and sets:

```

% Lattice operations
\newcommand{\jj}{\TextOrMath{$\vee$\xspace}{\vee}}
\newcommand{\mm}{\TextOrMath{$\wedge$\xspace}{\wedge}}
\newcommand{\JJ}{\bigvee}% big join
\newcommand{\MM}{\bigwedge}% big meet
\newcommand{\JJm}[2]{\JJ(\,,#1\mid#2\,,)}% big join with a middle
\newcommand{\MMm}[2]{\MM(\,,#1\mid#2\,,)}% big meet with a middle

% Set operations
\newcommand{\uu}{\cup}% union
\newcommand{\ii}{\cap}% intersection
\newcommand{\UU}{\bigcup}% big union
\newcommand{\II}{\bigcap}% big intersection
\newcommand{\UUm}[2]{\UU(\,,#1\mid#2\,,)}% big union with a middle
\newcommand{\IIIm}[2]{\II(\,,#1\mid#2\,,)}% big intersection with a middle

% Sets
\newcommand{\ci}{\subseteq}% contained in with equality
\newcommand{\nc}{\nsubseteq}% not \ci
\newcommand{\sci}{\subset}% strictly contained in
\newcommand{\nci}{\not\subset}% not \ci
\newcommand{\ce}{\supseteq}% containing with equality
\newcommand{\nce}{\nsupseteq}% not \ce
\newcommand{\nin}{\notin}% not \in

```



```

\newcommand{\es}{\varnothing}% the empty set
\newcommand{\set}[1]{\{#1\}}% set
\newcommand{\setm}[2]{\{ \, #1 \mid #2 \, \}}% set with a middle
\def\v v<#1>{\langle #1 \rangle}% vector

% Partial ordering
\newcommand{\nle}{\nleq}% not \leq

```

So `$a \jj b$` produces $a \vee b$ and `$A \contd B$` produces $A \subseteq B$, and so on.



Tip The original commands are not redefined, so if a coauthor would rather use `$a \vee b$` and not `$a \jj b$`, the `\vee` command is still available.

The commands with a “middle” are exemplified by `\setm`:

```
$\setm{x \in R}{x^2 \leq 2}$
```

which typesets as $\{x \in R \mid x^2 \leq 2\}$.

Using the `\set` command, we can type the set $\{a, b\}$ as `$\set{a,b}$`, which is easier to read than `$\{a,b\}$`. Similarly, we type `$\vect{a,b}$` for the vector $\langle a, b \rangle$, so it looks like a vector.

Next in `newlattice.sty`, I map the Greek letters and bold Greek letters to easy to remember commands. For some, I prefer to use the variants, but that is a matter of individual taste. It is also a matter of taste whether or not to change the commands for the Greek letters at all, and how far one should go in changing commonly used commands.

```

% Greek letters
\newcommand{\ga}{\TextOrMath{$\alpha$ \xspace}{\alpha}}
\newcommand{\gb}{\TextOrMath{$\beta$ \xspace}{\beta}}
\newcommand{\gc}{\TextOrMath{$\chi$ \xspace}{\chi}}
\newcommand{\gd}{\TextOrMath{$\delta$ \xspace}{\delta}}
\renewcommand{\ge}{\TextOrMath{$\varepsilon$ \xspace}{\varepsilon}}
\newcommand{\gf}{\TextOrMath{$\varphi$ \xspace}{\varphi}}
\renewcommand{\gg}{\TextOrMath{$\gamma$ \xspace}{\gamma}}
\newcommand{\gh}{\TextOrMath{$\eta$ \xspace}{\eta}}
\newcommand{\gi}{\TextOrMath{$\iota$ \xspace}{\iota}}
\newcommand{\gk}{\TextOrMath{$\kappa$ \xspace}{\kappa}}
\newcommand{\gl}{\TextOrMath{$\lambda$ \xspace}{\lambda}}
\newcommand{\gm}{\TextOrMath{$\mu$ \xspace}{\mu}}
\newcommand{\gn}{\TextOrMath{$\nu$ \xspace}{\nu}}
\newcommand{\go}{\TextOrMath{$\omega$ \xspace}{\omega}}

```

```

\newcommand{\gp}{\TextOrMath{$\pi$\xspace}{\pi}}
\newcommand{\gq}{\TextOrMath{$\theta$\xspace}{\theta}}
\newcommand{\gr}{\TextOrMath{$\varrho$\xspace}{\varrho}}
\newcommand{\gs}{\TextOrMath{$\sigma$\xspace}{\sigma}}
\newcommand{\gt}{\TextOrMath{$\tau$\xspace}{\tau}}
\newcommand{\gu}{\TextOrMath{$\upsilon$\xspace}{\upsilon}}
\newcommand{\gv}{\TextOrMath{$\vartheta$\xspace}{\vartheta}}
\newcommand{\gx}{\TextOrMath{$\xi$\xspace}{\xi}}
\newcommand{\gy}{\TextOrMath{$\psi$\xspace}{\psi}}
\newcommand{\gz}{\TextOrMath{$\zeta$\xspace}{\zeta}}

\newcommand{\gG}{\TextOrMath{$\Gamma$\xspace}{\Gamma}}
\newcommand{\gD}{\TextOrMath{$\Delta$\xspace}{\Delta}}
\newcommand{\gF}{\TextOrMath{$\Phi$\xspace}{\Phi}}
\newcommand{\gL}{\TextOrMath{$\Lambda$\xspace}{\Lambda}}
\newcommand{\gO}{\TextOrMath{$\Omega$\xspace}{\Omega}}
\newcommand{\gP}{\TextOrMath{$\Pi$\xspace}{\Pi}}
\newcommand{\gQ}{\TextOrMath{$\Theta$\xspace}{\Theta}}
\newcommand{\gS}{\TextOrMath{$\Sigma$\xspace}{\Sigma}}
\newcommand{\gU}{\TextOrMath{$\Upsilon$\xspace}{\Upsilon}}
\newcommand{\gX}{\TextOrMath{$\Xi$\xspace}{\Xi}}
\newcommand{\gY}{\TextOrMath{$\Psi$\xspace}{\Psi}}

% Bold Greek letters
\newcommand{\bga}{\TextOrMath{$\boldsymbol{\alpha}$\xspace}{\boldsymbol{\alpha}}}
\newcommand{\bgb}{\TextOrMath{$\boldsymbol{\beta}$\xspace}{\boldsymbol{\beta}}}
\newcommand{\bgc}{\TextOrMath{$\boldsymbol{\chi}$\xspace}{\boldsymbol{\chi}}}
\newcommand{\bgd}{\TextOrMath{$\boldsymbol{\delta}$\xspace}{\boldsymbol{\delta}}}
\newcommand{\bge}{\TextOrMath{$\boldsymbol{\varepsilon}$\xspace}{\boldsymbol{\varepsilon}}}
\newcommand{\bgf}{\TextOrMath{$\boldsymbol{\varphi}$\xspace}{\boldsymbol{\varphi}}}
\newcommand{\bgg}{\TextOrMath{$\boldsymbol{\gamma}$\xspace}{\boldsymbol{\gamma}}}
\newcommand{\bgh}{\TextOrMath{$\boldsymbol{\eta}$\xspace}{\boldsymbol{\eta}}}
\newcommand{\bgi}{\TextOrMath{$\boldsymbol{\iota}$\xspace}{\boldsymbol{\iota}}}

```

```

\newcommand{\bkg}{\TextOrMath{$\boldsymbol{\kappa}$\xspace}
  {\boldsymbol{\kappa}}}
\newcommand{\bgl}{\TextOrMath{$\boldsymbol{\lambda}$\xspace}
  {\boldsymbol{\lambda}}}
\newcommand{\bgm}{\TextOrMath{$\boldsymbol{\mu}$\xspace}
  {\boldsymbol{\mu}}}
\newcommand{\bgn}{\TextOrMath{$\boldsymbol{\nu}$\xspace}
  {\boldsymbol{\nu}}}
\newcommand{\bgo}{\TextOrMath{$\boldsymbol{\omega}$\xspace}
  {\boldsymbol{\omega}}}
\newcommand{\bgp}{\TextOrMath{$\boldsymbol{\pi}$\xspace}
  {\boldsymbol{\pi}}}
\newcommand{\bgq}{\TextOrMath{$\boldsymbol{\theta}$\xspace}
  {\boldsymbol{\theta}}}
\newcommand{\bgr}{\TextOrMath{$\boldsymbol{\varrho}$\xspace}
  {\boldsymbol{\varrho}}}
\newcommand{\bgs}{\TextOrMath{$\boldsymbol{\sigma}$\xspace}
  {\boldsymbol{\sigma}}}
\newcommand{\bgt}{\TextOrMath{$\boldsymbol{\tau}$\xspace}
  {\boldsymbol{\tau}}}
\newcommand{\bgu}{\TextOrMath{$\boldsymbol{\upsilon}$\xspace}
  {\boldsymbol{\upsilon}}}
\newcommand{\bgv}{\TextOrMath{$\boldsymbol{\vartheta}$\xspace}
  {\boldsymbol{\vartheta}}}
\newcommand{\bgx}{\TextOrMath{$\boldsymbol{\xi}$\xspace}
  {\boldsymbol{\xi}}}
\newcommand{\bgy}{\TextOrMath{$\boldsymbol{\psi}$\xspace}
  {\boldsymbol{\psi}}}
\newcommand{\bgz}{\TextOrMath{$\boldsymbol{\zeta}$\xspace}
  {\boldsymbol{\zeta}}}
\newcommand{\bgL}{\TextOrMath{$\boldsymbol{g_L}$\xspace}
  {\boldsymbol{g_L}}}
\newcommand{\bgF}{\TextOrMath{$\boldsymbol{g_F}$\xspace}
  {\boldsymbol{g_F}}}

```

I also introduce some new names for text font commands by abbreviating text to `t` (so that `\textbf` becomes `\tbf`) and for math font commands by abbreviating math to `m` (so that `\mathbf` becomes `\mbf`).

```
% Font commands
\newcommand{\tbfbf}{\textbf}% text bold
\newcommand{\tit}{\textit}% text italic
\newcommand{\tsl}{\textsl}% text slanted
\newcommand{\tsc}{\textsc}% text small cap
\newcommand{\ttt}{\texttt}% text typewriter
\newcommand{\trm}{\textrm}% text roman
\newcommand{\tsf}{\textsf}% text sans serif
\newcommand{\tup}{\textup}% text upright

\newcommand{\mbf}{\mathbf}% math bold
\providecommand{\mit}{\mathit}% math italic
\newcommand{\msf}{\mathsf}% math sans serif
\newcommand{\mrm}{\mathrm}% math roman
\newcommand{\mtt}{\mathtt}% math typewriter
```

The math alphabets are invoked as commands with arguments: `\Bold` for bold, `\Cal` for calligraphic, `\DD` for blackboard bold (double), and `\Frak` for fraktur (German Gothic—see Section 8.4.2. Notice that `\Cal` and `\Euler` are different because of the option `mathscr` of the `eucal` package (see Section 8.4.1).

```
\newcommand{\B}{\boldsymbol}
% Bold math symbol, use as \B{a}
\newcommand{\C}[1]{\mathcal{#1}}
% Euler Script - only caps, use as \C{A}
\newcommand{\D}[1]{\mathbb{#1}}
% Doubled - blackboard bold - only caps, use as \D{A}
\newcommand{\E}[1]{\mathcal{#1}}% same as \C
% Euler Script - only caps, use as \E{A}
\newcommand{\F}[1]{\mathfrak{#1}}% Fraktur, use as \F{a}

%Sansserif, special lattices
%Chains and Boolean lattces
```

```
\newcommand{\SC}[1]{\msf{C}_{#1}}
\newcommand{\SB}[1]{\msf{B}_{#1}}
\newcommand{\SD}[1]{\msf{D}_{#1}}
\newcommand{\SL}[1]{\msf{L}_{#1}}
\newcommand{\SM}[1]{\msf{M}_{#1}}
\newcommand{\SMb}[1]{\msf{M}_{3}[#1]_ \text{bal}}
\newcommand{\SN}[1]{\msf{N}_{#1}}
\newcommand{\SH}[1]{\msf{H}_{#1}}
\newcommand{\SV}[1]{\msf{V}_{#1}}
```

```

\newcommand{\SfC}[1]{\msf{C}_{#1}}
\newcommand{\SfB}[1]{\msf{B}_{#1}}
\newcommand{\SfM}[1]{\msf{M}_{#1}}
\newcommand{\SfN}[1]{\msf{N}_{#1}}
\newcommand{\SfS}[1]{\msf{S}_{#1}}

```

Here are some commands of importance in lattice theory:

```

% Constructs
\DeclareMathOperator{\Id}{Id}
\DeclareMathOperator{\Fi}{Fi}_{l}
\DeclareMathOperator{\Con}{Con}
\DeclareMathOperator{\Aut}{Aut}
\DeclareMathOperator{\Sub}{Sub}
\DeclareMathOperator{\Pow}{Pow}
\DeclareMathOperator{\Part}{Part}
\DeclareMathOperator{\Ker}{Ker}
\DeclareMathOperator{\Joinir}{Join}
\DeclareMathOperator{\Meetir}{Meet}
\DeclareMathOperator{\Down}{Down}
\DeclareMathOperator{\Ji}{Ji}
\DeclareMathOperator{\Mi}{Mi}

% Generated by
\newcommand{\con}[1]{\tup{con}(#1)}
\newcommand{\consub}[2]{\tup{con}_{#1}(#2)}
\newcommand{\sub}[1]{\tup{sub}(#1)}
\newcommand{\id}[1]{\tup{id}(#1)}
\newcommand{\fil}[1]{\tup{f}{il}(#1)}

% Miscellaneous
\newcommand{\nl}{\newline}
\newcommand{\ol}[1]{\overline{#1}}
\newcommand{\ul}[1]{\underline{#1}}
\providecommand{\bysame}{\makebox[3em]{\hrulefill}\thinspace}
\newcommand{\q}{\quad}% spacing
\newcommand{\qq}{\quad\quad}% more spacing
\newcommand{\iso}{\cong}% isomorphic
\newenvironment{enumeratei}{\begin{enumerate}[\upshape (i)]}%
{\end{enumerate}}
%produces (i), (ii), etc. Cross-reference with \eqref.
\newenvironment{enumeratea}{\begin{enumerate}[\upshape (a)]}%
{\end{enumerate}}
%produces (a), (b), etc. Cross-reference with \eqref.

```

```

\theoremstyle{plain}
\newtheorem*{namedtheorem}{\theoremname}
\newcommand{\theoremname}{testing}
\newenvironment{named}[1]{\renewcommand{\theoremname}{#1}}
  \begin{namedtheorem}}
  {\end{namedtheorem}}
%use it as \begin{named}{Name of theorem}
%Body of theorem \end{named}

\newcommand{\Dg}{\downarrow\!}% down-set generated by congruences
\newcommand{\per}{\sim}% perspective ~
\newcommand{\pu}{\stackrel{\text{u}}{\sim}}
  % perspective up ~ with u on top
\newcommand{\pd}{\stackrel{\text{d}}{\sim}}
  % perspective down ~ with d on top
\newcommand{\proj}{\approx}% projective
\newcommand{\cpu}{\nearrow}
  % congruence perspective up -- up arrow
\newcommand{\cpd}{\searrow}
  % congruence perspective down -- down arrow
\newcommand{\cper}{\hookrightarrow}
  % congruence perspective onto-- hooked right arrow
\newcommand{\cproj}{\Rrightarrow}
  % congruence projective into -- double headed arrow
\newcommand{\cprojboth}{\Leftrightarrow}
  % congruence projective both ways
  %-- two headed double arrow

%perspective
\newcommand{\perspsymb}{\thicksim}% perspective symbol
\newcommand{\persp}{\perspsymb}% perspective
\newcommand{\perspup}{\stackrel{\text{up}}{\perspsymb}}
  % perspective up
\newcommand{\perspdn}{\stackrel{\text{dn}}{\perspsymb}}
  % perspective down

% c-perspective
\newcommand{\cperspsymb}{\hookrightarrow}
  % c-perspective symbol
\newcommand{\cpersp}{\cperspsymb}% c-perspective
\newcommand{\cperspup}{\stackrel{\text{up}}{\cperspsymb}}
  % c-perspective up

```

```

\newcommand{\cperspdn}{\stackrel{\text{rm}{dn}}{\cperspsymb}}
% c-perspective dn

\newcommand{\lp}{\tup{}}
\newcommand{\rp}{\tup{}}\xspace
\newcommand{\up}[1]{\tup{(#1\tup{)}}}
\newcommand{\one}{\mathbf{1}}
\newcommand{\zero}{\mathbf{0}}
\newcommand{\restr}{\rceil}

\newcommand{\itemref}[1]{\eqref{#1}}

\def\cng#1=#2(#3){#1\equiv#2\pmod{#3}}
%congruence, use it as \cng a=b(\theta)%
\def\cngd#1=#2(#3){#1\equiv#2!\pmod{#3}}
%congruence for display, use it as \cngd a=b(\theta)%
\def\ncng#1=#2(#3){#1\not\equiv#2\pmod{#3}}
%negate cng
\def\ncngd#1=#2(#3){#1\not\equiv#2!\pmod{#3}}
%negate dcng

\endinput

```

See Section 15.1.9 for the `\congr` command. The `enumeratei` and named environments are discussed in Section 15.2.1. The `enumeraea` environment is similar.

This command file, like all command files, is terminated with the `\endinput` command.

Your style file will evolve with time.



Tip Keep a copy of your style file in the folder of an article you write. This way, years later, even though your style file has changed, you can typeset the original article with no problem.

The `\TextOrMath` command is very useful. For instance, we can use it to define our Greek letters, such as

```
\newcommand{\ga}{\TextOrMath{\alpha}}{\alpha}}
```

Then we can use `\ga` both in text and math to produce α .

15.4 The sample article with custom commands

In this section, we look at the `art2ccom.tex` sample article, which is a rewrite of the `art2.tex` sample article utilizing the custom commands collected in the command file `newlattice.sty` (for these files, see Section 15.3 and the `samples` folder).

```

% Sample file: art2ccom.tex
% The sample article with custom commands and environments

\documentclass{amsart}
\usepackage{newlattice}

\theoremstyle{plain}
\newtheorem{theorem}{Theorem}
\newtheorem{corollary}{Corollary}
\newtheorem{lemma}{Lemma}
\newtheorem{proposition}{Proposition}

\theoremstyle{definition}
\newtheorem{definition}{Definition}

\theoremstyle{remark}
\newtheorem*{notation}{Notation}

\numberwithin{equation}{section}

\newcommand{\Prodm}[2]{\gP(\,#1\mid#2\,)}
% product with a middle
\newcommand{\Prodsm}[2]{\gP^{*}(\,#1\mid#2\,)}
% product * with a middle
\newcommand{\vectsup}[2]{\vect<\dots,0,\dots,\overset{#1}{#2},%
\dots,0,\dots>}% special vector
\newcommand{\Dsqr}[2]{D^{\langle#1\rangle}}

\begin{document}
\title[Complete-simple distributive lattices]
{A construction of complete-simple\
distributive lattices}
\author{George~A. Menuhin}
\address{Computer Science Department\
University of Winnebago\
Winnebago, Minnesota 23714}
\email{menuhin@ccw.uwinnebago.edu}
\urladdr{http://math.uwinnebago.edu/homepages/menuhin/}
\thanks{Research supported by the NSF under grant number~23466.}
\keywords{Complete lattice, distributive lattice, complete
congruence, congruence lattice}
\subjclass[2020]{Primary: 06B10; Secondary: 06D05}

```



```

\date{March 15, 2024}

\begin{abstract}
  In this note we prove that there exist \emph{complete-simple
    distributive lattices,} that is, complete distributive
    lattices in which there are only two complete congruences.
\end{abstract}
\maketitle

\section{Introduction}\label{S:intro}
In this note we prove the following result:

\begin{named}{Main Theorem}
  There exists an infinite complete distributive lattice
   $K$  with only the two trivial complete congruence relations.
\end{named}

\section{The  $\mathcal{Dsq}$  construction}\label{S:Ds}
For the basic notation in lattice theory and universal algebra,
see Ferenc~R. Richardson~\cite{fR23} and George~A.
Menuhin~\cite{gM18}. We start with some definitions:

\begin{definition}\label{D:prime}
  Let  $V$  be a complete lattice, and let  $F[p] = [u, v]$  be
  an interval of  $V$ . Then  $F[p]$  is called
  \emph{complete-prime} if the following three conditions
  are satisfied:
  \begin{enumeratei}
    \item  $u$  is meet-irreducible but  $u$  is \emph{not}
      completely meet-irreducible;\label{m-i}
    \item  $v$  is join-irreducible but  $v$  is \emph{not}
      completely join-irreducible;\label{j-i}
    \item  $[u, v]$  is a complete-simple lattice.\label{c-s}
  \end{enumeratei}
\end{definition}

Now we prove the following result:

\begin{lemma}\label{L:Dsq}
  Let  $D$  be a complete distributive lattice satisfying
  conditions \itemref{m-i} and~\itemref{j-i}.
  Then  $\mathcal{Dsq}$  is a sublattice of  $D^2$ ; hence  $\mathcal{Dsq}$  is

```

a lattice, and \mathcal{D} is a complete distributive lattice satisfying conditions \itemref{m-i} and \itemref{j-i}.

\end{lemma}

\begin{proof}

By conditions \itemref{m-i} and \itemref{j-i}, \mathcal{D} is a sublattice of \mathcal{D}^2 . Hence, \mathcal{D} is a lattice.

Since \mathcal{D} is a sublattice of a distributive lattice, \mathcal{D} is a distributive lattice. Using the characterization of standard ideals in Ernest~T. Moynahan~\cite{eM01}, \mathcal{D} has a zero and a unit element, namely, $\langle 0, 0 \rangle$ and $\langle 1, 1 \rangle$. To show that \mathcal{D} is complete, let $a \in A \subseteq \mathcal{D}$, and let $a = \bigvee A$ in \mathcal{D}^2 . If $a \in \mathcal{D}$, then $a = \bigvee A$ in \mathcal{D} ; otherwise, a is of the form $\langle b, 1 \rangle$ for some $b \in D$ with $b < 1$. Now $\bigvee A = \langle 1, 1 \rangle$ in \mathcal{D}^2 , and the dual argument shows that $\bigwedge A$ also exists in \mathcal{D}^2 . Hence \mathcal{D} is complete. Conditions \itemref{m-i} and \itemref{j-i} are obvious for \mathcal{D} .

\end{proof}

\begin{corollary}\label{C:prime}

If \mathcal{D} is complete-prime, then so is \mathcal{D} .

\end{corollary}

The motivation for the following result comes from Soo-Key Foo~\cite{sF10}.

\begin{lemma}\label{L:ccr}

Let \mathcal{g} be a complete congruence relation of \mathcal{D} such that

$$\langle 1, d \rangle \mathcal{g} \langle 1, 1 \rangle \text{ for some } d \in D \text{ with } d < 1. \text{ Then } \mathcal{g} = \mathcal{g}_i.$$

\end{lemma}

\begin{proof}

Let \mathcal{g} be a complete congruence relation of \mathcal{D} satisfying \itemref{E:rigid}. Then $\mathcal{g} = \mathcal{g}_i$.

\end{proof}

`\section{The $\mathcal{G}P^*$ construction}\label{S:P*}`

The following construction is crucial to our proof of the Main~Theorem:

`\begin{definition}\label{D:P*}`

Let $D_{\{i\}}$, for $i \in I$, be complete distributive lattices satisfying condition~\itemref{j-i}. Their $\mathcal{G}P^*$ product is defined as follows:

`\[`

$$\mathcal{P}rodsm{D_{\{i\}}}{i \in I} = \mathcal{P}rodsm{D_{\{i\}}^{\{-\}}}{i \in I} + 1;$$

`\]`

that is, $\mathcal{P}rodsm{D_{\{i\}}}{i \in I}$ is

$\mathcal{P}rodsm{D_{\{i\}}^{\{-\}}}{i \in I}$ with a new unit element.

`\end{definition}`

`\begin{notation}`

If $i \in I$ and $d \in D_{\{i\}}^{\{-\}}$, then

`\[`

$$\mathcal{V}ectsup{i}{d}$$

`\]`

is the element of $\mathcal{P}rodsm{D_{\{i\}}}{i \in I}$ whose i -th component is d and all the other components are 0 .

`\end{notation}`

See also Ernest~T. Moynahan~\cite{eM01}. Next we verify:

`\begin{theorem}\label{T:P*}`

Let $D_{\{i\}}$, for $i \in I$, be complete distributive lattices satisfying condition~\itemref{j-i}. Let $\mathcal{G}Q$ be a complete congruence relation on

$\mathcal{P}rodsm{D_{\{i\}}}{i \in I}$. If there exist

$i \in I$ and $d \in D_{\{i\}}$ with $d < 1_{\{i\}}$ such

that for all $d \leq c < 1_{\{i\}}$,

`\begin{equation}\label{E:cong1}`

$$\mathcal{C}ng\mathcal{V}ectsup{i}{d} = \mathcal{V}ectsup{i}{c}(\mathcal{G}Q),$$

`\end{equation}`

then $\mathcal{G}Q = \mathcal{G}I$.

`\end{theorem}`

`\begin{proof}`

Since

```
\begin{equation}\label{E:cong2}
  \cng{\vectsup{i}{d}}{\vectsup{i}{c}}(\gQ),
\end{equation}
and  $\gQ$  is a complete congruence relation, it follows
from condition~\itemref{c-s} that
\begin{equation}\label{E:cong}
\begin{split}
&\langle \dots, \overset{i}{d}, \dots, 0, \\
&\quad \dots \rangle \\
&\equiv \bigvee ( \langle \dots, 0, \dots, \\
&\quad \overset{i}{c}, \dots, 0, \dots \rangle \mid d \leq c < 1) \\
&\equiv 1 \pmod{\Theta}.
\end{split}
\end{equation}
```

Let $j \in I$, for $j \neq i$, and let
 $a \in D_{\{j\}^{\{-\}}}$. Meeting both sides of the congruence
\itemref{E:cong} with $\vectsup{j}{a}$, we obtain

```
\begin{equation}\label{E:comp}
  \begin{split}
    0 \equiv \vectsup{i}{d} \mm \vectsup{j}{a} \\
    \equiv \vectsup{j}{a} \pod{\gQ}.
  \end{split}
\end{equation}
```

Using the completeness of \gQ and \itemref{E:comp}, we get:

```
\begin{equation}\label{E:cong3}
  \cng{0=\JJm{\vectsup{j}{a}}{a \in D_{\{j\}^{\{-\}}}}}{1}(\gQ),
\end{equation}
hence  $\gQ = \gi$ .
```

\end{proof}

```
\begin{theorem}\label{T:P*a}
  Let  $D_{\{i\}}$ , for  $i \in I$ , be complete distributive
  lattices satisfying
  conditions \itemref{j-i} and~\itemref{c-s}. Then
   $\Prods{D_{\{i\}}}{i \in I}$  also satisfies
  conditions~\itemref{j-i} and \itemref{c-s}.
\end{theorem}
```

\begin{proof}

Let \gQ be a complete congruence on

```

 $\prod_{D_i} \{i \in I\}$ . Let  $i \in I$ . Define
\begin{equation}\label{E:dihat}
\widehat{D}_i = \text{setm} \{ \text{vectsup}_i[d] \mid d \in D_i \}
\text{uu} \text{set}\{1\}.
\end{equation}
Then  $\widehat{D}_i$  is a complete sublattice of
 $\prod_{D_i} \{i \in I\}$ , and  $\widehat{D}_i$ 
is isomorphic to  $D_i$ . Let  $gQ_i$  be the
restriction of  $gQ$  to  $\widehat{D}_i$ . Since
 $D_i$  is complete-simple, so is  $\widehat{D}_i$ ,
hence  $gQ_i$  is  $g_o$  or  $g_i$ . If  $gQ_i = g_o$ 
for all  $i \in I$ , then  $gQ = g_o$ .
If there is an  $i \in I$ , such that  $gQ_i = g_i$ ,
then  $\text{cng}_0 = 1(gQ)$ , and hence  $gQ = g_i$ .
\end{proof}

```

The Main Theorem follows easily from Theorems~\ref{T:P*} and \ref{T:P*a}.

```
\begin{thebibliography}{9}
```

```
\bibitem{sF10}
```

Soo-Key Foo, *\emph{Lattice Constructions}*, Ph.D. thesis,
University of Winnebago, Winnebago, MN, December, 2010.

```
\bibitem{gM18}
```

George~A. Menuhin, *\emph{Universal algebra}*. D.~van
Nostrand, Princeton, 2018.

```
\bibitem{eM01}
```

Ernest~T. Moynahan, *\emph{On a problem of M. Stone}*,
Acta Math. Acad. Sci. Hungar. \textbf{8} (2001), 455--460.

```
\bibitem{eM19}
```

\bysame, *\emph{Ideals and congruence relations in
lattices}*.~II, *Magyar Tud. Akad. Mat. Fiz. Oszt. K\~{o}zl.*
\textbf{9} (2019), 417--434 (Hungarian).

```
\bibitem{fR23}
```

Ferenc~R. Richardson, *\emph{General lattice theory}*. Mir,
Moscow, expanded and revised ed., 2023 (Russian).

```
\end{thebibliography}
\end{document}
```

15.5 Numbering and measuring

L^AT_EX stores integers in *counters*. For example, the section counter contains the current section number. Distance measurements are saved in *length commands*. For instance, the `\textwidth` command contains the width of the text. For this book, the length command `\textwidth` is set to 345.0 points.

In this section, we take a closer look at counters and length commands.

15.5.1 Counters

Counters may be defined by L^AT_EX, by document classes, by packages, or by the user.

Standard L^AT_EX counters

L^AT_EX automatically generates numbers for equations, sections, theorems, and so on. Each such number is stored in a *counter*. Table 15.2 shows the standard L^AT_EX counters. Their names are more or less self-explanatory. In addition, for every proclamation *name*, there is a matching counter called `name` (see Section 4.4).

equation	part	enumi
figure	chapter	enumii
footnote	section	enumiii
mpfootnote	subsection	enumiv
table	subsubsection	
pages	paragraph	
totalpages	subparagraph	

Table 15.2: Standard L^AT_EX counters.

Setting counters

The command for setting a counter's value is `\setcounter`. When L^AT_EX generates a number, it first increments the appropriate counter, so if you want the next chapter to be numbered 3, you should set the chapter counter to 2 by typing

```
\setcounter{chapter}{2}
```

before the `\chapter` command. The only exception to this rule is the page number, which is first used to number the current page, and then incremented. If you wanted to set the current page number to 63, you would include the command

```
\setcounter{page}{63}
```

somewhere in the page.

L^AT_EX initializes and increments its standard counters automatically. Sometimes you may want to manipulate them yourself. To typeset only `chapter3.tex`, the third chapter of your book, start with

```
\setcounter{chapter}{2}
\include{chapter3}
```

and when `chapter3.tex` is typeset, the chapter is properly numbered. You can also type

```
\setcounter{page}{63}
```

if the first page of this chapter is supposed to be 63.



Tip If you need to manipulate counters, always look for solutions in which L^AT_EX does the work for you.

Defining new counters

You can define your own counters. For example,

```
\newcounter{mycounter}
```

makes `mycounter` a new counter. In the definition, you can use an optional argument, the name of another counter:

```
\newcounter{mycounter}[basecounter]
```

which automatically resets `mycounter` to 0 if `basecounter` changes value. This command has the same form as the command L^AT_EX uses internally for tasks such as numbering theorems and subsections within sections.



Rule: New counters

New counters have to be defined in the preamble of the document. They should not be defined in a file read in with an `\include` command.

Let us suppose that you define a new counter, `mycounter`, in `chapter5.tex`, which is made part of your whole document with an `\include` command. When you typeset your document with `\includeonly` commands not including `chapter5.tex`, you get a message, such as

```
! LaTeX Error: No counter 'mycounter' defined.
```

Style	Command	Sample
Arabic	<code>\arabic{counter}</code>	1, 2, ...
Lowercase Roman	<code>\roman{counter}</code>	i, ii, ...
Uppercase Roman	<code>\Roman{counter}</code>	I, II, ...
Lowercase Letters	<code>\alph{counter}</code>	a, b, ..., z
Uppercase Letters	<code>\Alph{counter}</code>	A, B, ..., Z

Table 15.3: Counter styles.

Counter styles

The value of `counter` can be displayed in the typeset document with the command

```
\thecounter
```

If you want to change the counter's appearance when typeset, issue the command

```
\renewcommand{\thecounter}{new_style}
```

where *new_style* specifies the counter modified as shown in Table 15.3. The default style is arabic. For instance, if you give the command

```
\renewcommand{\thetheorem}{\Alph{theorem}}
```

then the theorems appear as **Theorem A**, **Theorem B**, ...

Here is a more complicated example for a book:

```
\renewcommand{\thechapter}{\arabic{chapter}}
\renewcommand{\thesection}{\thechapter-\arabic{section}}
\renewcommand{\thesubsection}
{\thechapter-\arabic{section}.\arabic{subsection}}
```

With these definitions, Section 1 of Chapter 3 is numbered in the form 3-1 and Subsection 2 of Section 1 of Chapter 3 is numbered in the form 3-1.2.

The `\pagenumbering` command is a shorthand method for setting the page numbering in a given style. For instance, `\pagenumbering{roman}` numbers pages as i, ii, and so on.

The `subequations` environment (see Section 8.6) uses `parentequation` as the counter for the whole equation group and it uses `equation` as the counter for the subequations. To change the default format of the equation numbers from (2a), (2b), and so forth, to (2i), (2ii), and so on, type the following line inside the `subequations` environment

```
\renewcommand{\theequation}
{\theparentequation\roman{equation}}
```


If you want equation numbers like (2.i), (2.ii), and so on, type

```
\renewcommand{\theequation}%
      {\theparentequation.\roman{equation}}
```

Counter arithmetic

The `\stepcounter{counter}` command increments `counter` and sets all the counters that were defined with the optional argument `counter` to 0. The variant

```
\refstepcounter{counter}
```

does the same, and also sets the value for the next `\label` command.

You can do some arithmetic with the command

```
\addtocounter{counter}{n}
```

where n is an integer. For example,

```
\setcounter{counter}{5}
\addtocounter{counter}{2}
```

sets `counter` to 7.

The value stored in a counter can be accessed using the `\value` command, which is mostly used with the `\setcounter` or `\addtocounter` commands. For instance, you can set `counter` to equal the value of another counter, `oldcounter`, by typing

```
\setcounter{counter}{\value{oldcounter}}
```

Here is a typical example of counter manipulation. You have a theorem (invoked in a `theorem` environment) and you want it followed by several corollaries (each in a `corollary` environment) starting with Corollary 1. In other words, Theorem 1 should be followed by Corollary 1, Corollary 2, and then Theorem 2. By default, \LaTeX numbers the next corollary as Corollary 3, even if it follows another theorem. To tell \LaTeX to start numbering the corollaries from 1 again, issue the command

```
\setcounter{corollary}{0}
```

after each theorem. But such a process is error-prone, and goes against the spirit of \LaTeX .

Instead, follow my advice on page 444, and let \LaTeX do the work for you. In the preamble, type the proclamations

```
\newtheorem{theorem}{Theorem}
\newtheorem{corollary}{Corollary}[theorem]
```

We are almost there. Theorem 1 now is followed by Corollary 1.1, Corollary 1.2 and Theorem 3 by Corollary 3.1. If we redefine `\thecorollary`,

```
\renewcommand{\thecorollary}{\arabic{corollary}}
```

then Theorem 1 is followed by Corollary 1 and Corollary 2, and Theorem 3 is also followed by Corollary 1.

If you need to perform more complicated arithmetic with counters, use Kresten K. Thorup and Frank Jensen’s `calc` package. This package is discussed in Section A.5.2 of LC3.

Two special counters

The `secnumdepth` and `tocdepth` counters control which sectional units are numbered and which are listed in the table of contents, respectively. For example,

```
\setcounter{secnumdepth}{2}
```

sets `secnumdepth` to 2. As a result, chapters—if they are present in the document class—sections, and subsections are numbered, but subsubsections are not. This command must be placed in the preamble of the document. `tocdepth` is similar.

15.5.2 Length commands

While a counter contains integers, a length command contains a *real number* and a *dimensional unit*.

L^AT_EX recognizes many different dimensional units. We list five *absolute* units:

- `cm` centimeter,
- `in` inch,
- `pc` pica (1 `pc` = 12 `pt`),
- `pt` point (1 `in` = 72.27 `pt`),
- `mm` millimeter,

and two *relative* units:

- `em`, approximately the width of the letter M in the current font;
- `ex`, approximately the height of the letter x in the current font.

L^AT_EX defines many length commands. For instance, Section 5.1 of LC3 lists 17 length commands for page layout alone. You can find some of them in Figure 10.4. A list environment sets about a dozen additional length commands (see Figure 15.2). Length commands are defined for almost every aspect of L^AT_EX’s work, including displayed math environments—a complete list would probably contain a few hundred. Many are listed in Leslie Lamport’s [32] and many more in LC3. Others are hidden in packages such as `amsmath`.

The most common length commands are:

- `\parindent`, the amount of indentation at the beginning of a paragraph;
- `\parskip`, the extra vertical space inserted between paragraphs;
- `\textwidth`, the width of the text on a page.

A more esoteric example is `\marginparpush`, the minimum vertical space between two marginal notes. Luckily, you do not have to be familiar with many length commands because \LaTeX and the document class set them for you.

Defining new length commands

You can define your own length commands. For example,

```
\newlength{\mylength}
```

makes `\mylength` a new length command with a value of 0 points. Note that while you have to type

```
\newcounter{counter}
```

to get a new counter, typing

```
\newlength{\mylength}
```

results in a message such as

```
! Missing control sequence inserted.
<inserted text>
          \inaccessible
1.3 \newlength{\mylength}
```

Setting length

The `\setlength` command sets or resets the value of a length command. So

```
\setlength{\textwidth}{3in}
```

creates a very narrow page. The first argument of `\setlength` must be a length command, not simply the command name, that is

```
\setlength{textwidth}{3in} % Bad
```

is incorrect. The second argument of `\setlength` must be a real number with a dimensional unit, for instance, `3in`, and *not simply a real number*. In other words,

```
\setlength{\textwidth}{3} % Bad
```

is also incorrect. You can also use `\setlength` as an environment, as in Section 9.3.



Tip A common mistake is to type a command such as

```
\setlength{\marginpar}{0}
```

Instead, type

```
\setlength{\marginpar}{0pt}
```

Always be sure to include a dimensional unit.

The `\addtolength` command adds a quantity to the value of a length command. For instance,

```
\addtolength{\textwidth}{-10pt}
```

narrows the page width by 10 points.

If you define

```
\newlength{\shorterlength}
\setlength{\shorterlength}{\mylength}
\addtolength{\shorterlength}{-.5in}
```

then `\parbox{\shorterlength}{...}` always typesets its second argument in a box 1/2 inch narrower than the parboxes set to be of width `\mylength`.

When \LaTeX typesets some text or math, it creates a box. Three measurements are used to describe the size of the box:

- the width,
- the height, from the baseline to the top,
- the depth, from the baseline to the bottom,

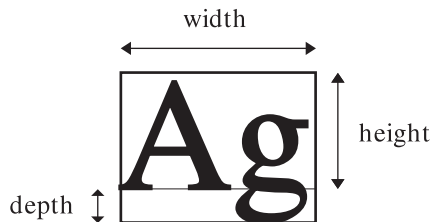


Figure 15.1: The measurements of a box.

as illustrated in Figure 15.1. For instance, the box typesetting “aa” has a width of 10.00003 pt, a height of 4.30554 pt, and a depth of 0 pt. The box typesetting “ag” has the same width and height, but a depth of 1.94444 pt. The box “Ag” (see Figure 15.1) has a width of 12.50003 pt, a height of 6.83331 pt, and a depth of 1.94444 pt.

The commands

```
\settowidth
\settoheight
\settodepth
```

each take two arguments. The first argument is a length command, the second is text (or math) to be measured by \LaTeX . The corresponding measurement of the box in which the second argument is typeset is assigned to the length command in the first argument. For example, if `\mylength` is a length command, then

```
\settowidth{\mylength}{Ag}
```

assigns 12.50003 pt to `\mylength`. It should be clear from this example how the `\phantom` and `\hphantom` commands (see Section 3.7.1) are related to this command.

To perform more complicated arithmetic with length commands, use the `calc` package.

Rubber lengths

In addition to rigid lengths, such as 3in, \LaTeX can also set *rubber lengths*, that is, lengths that are allowed to stretch and shrink. Here is an example:

```
\setlength{\stretchspace}{3in plus 10pt minus 8pt}
```

Assuming that `\stretchspace` is a length command, this command assigns it a value of 3 inches that can stretch by 10 points or shrink by 8 points, if necessary. So a box of width `\stretchspace` is 3 inches wide, plus up to 10 points, or minus up to 8 points.

Stretchable vertical spaces are often used before and after displayed text environments. \LaTeX adjusts these spaces to make the page look balanced. An example can be found in Section 15.1.8. `\medskipamount` is defined as

```
6.0pt plus 2.0pt minus 2.0pt
```

See Section 15.6.3 for more examples.

The `\fill` command is a special rubber length that can stretch any amount. The stretching is done evenly if there is more than one `\fill` present. See the second example of brand-new environments in Section 15.2.5.

15.6 Custom lists

Although there are three ready-made list environments provided by \LaTeX (see Section 4.2), it is often necessary to create one of your own using \LaTeX 's `list` environment. In fact, \LaTeX itself uses the `list` environment to define many of its standard environments, including:

- the three list environments (Section 4.2);
- the `quote`, `quotation`, and `verse` environments (Section 4.8);
- proclamations (Section 4.4);
- the style environments `center`, `flushleft`, and `flushright` (Section 4.3);
- the `thebibliography` environment (Section 10.5.1);
- the `theindex` environment (Section 10.5.2).

15.6.1 Length commands for the `list` environment

The general layout of a list is shown in Figure 15.2. It uses six horizontal measurements and three vertical measurements. I now list these length commands.

Vertical length commands

`\topsep` is most of the vertical space between the first item and the preceding text, and also between the last item and the following text. This space also includes `\parskip`, the extra vertical space inserted between paragraphs, and optionally, `\partopsep`, provided that the list environment starts a new paragraph.

`\parsep` is the space between paragraphs of the same item.

`\itemsep` is the space between items. Like `\topsep`, the actual gap is the sum of `\itemsep` and `\parsep`.

All of these vertical length commands are rubber lengths (see Section 15.5.2).

Horizontal length commands

By default, the margins of a `list` environment are the same as the margins of the surrounding text. If the list is nested within a list, the margins are wider and so the text is narrower.

The `\leftmargin` and `\rightmargin` length commands specify the distance between the edge of the item box and the left and right margins of the page.

The `label` is the text provided by the optional argument of an `\item` command or provided as a default in the definition of the list environment. It is typeset in a box

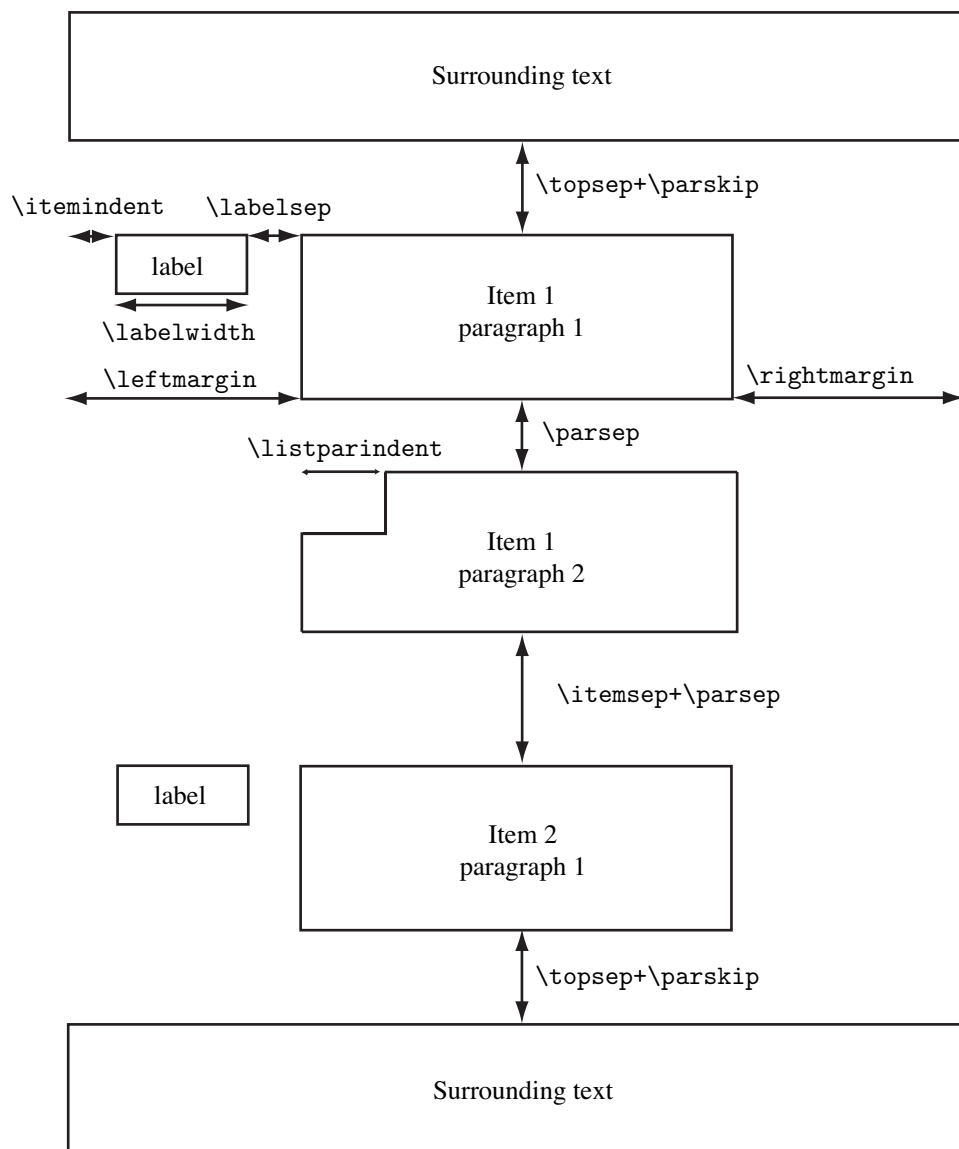


Figure 15.2: The layout of a custom list.

of width `\labelwidth`, which is indented `\itemindent` units from the left margin, and separated by a space of `\labelsep` units from the text box. If the label is too wide to fit in the box, it is typeset at its full natural width, and the first line in the text box is indented.

The second and subsequent paragraphs of an item are typeset with their first lines indented by `\listparindent` units.

15.6.2 The list environment

Custom lists are created with the `list` environment, which is invoked as follows:

```
\begin{list}{default_label}{declarations}
  \item item1
  \item item2
  ...
\end{list}
```

The arguments are:

- *default_label*, the label for any items that do not specify their own, similar to the optional argument of the `\item` command;
- *declarations*, the vertical and horizontal length commands and any other required parameters for the list.

Here is a very simple example:

┌

Here are the most important L^AT_EX rules about spaces in text, sentences, and paragraphs:

- ◊ **Rule 1:** Two or more spaces in text are the same as one.
- ◊ **Rule 2:** A blank line (that is, two end-of-line characters separated only by blanks and tabs) indicates the end of a paragraph.

Rules 1 and 2 make typing and copying very convenient.

└

I have used the ◊ math symbol (`\diamondsuit`) as a default label, and I set the item box 0.5 inch from either margin. So this example is typed as follows:

```
\noindent Here are the most important \la\ rules about
spaces in text, sentences, and paragraphs:
\begin{list}{\diamondsuit}{\setlength{\leftmargin}%
  {.5in}\setlength{\rightmargin}{.5in}}
\item \textbf{Rule 1:} Two or more spaces in text
are the same as one.
```



```
\item \textbf{Rule 2:} A blank line (that is, two
end-of-line characters separated only by blanks and tabs)
indicates the end of a paragraph.
\end{list}
Rules 1 and~2 make typing and copying very convenient.
```

Here is a second variant:

┌

Here are the most important L^AT_EX rules about spaces in text, sentences, and paragraphs:

Rule 1: Two or more consecutive spaces in text are the same as one.

Rule 2: A blank line (that is, two end-of-line characters separated only by blanks and tabs) indicates the end of a paragraph.

└

Rules 1 and 2 make typing and copying very convenient.

In this example, I dropped the optional *default_label* and typed **Rule 1:** and **Rule 2:** as (optional) arguments of the `\item` commands:

```
\noindent Here are the most important \la\ rules about
spaces in text, sentences, and paragraphs:
\begin{list}{}{\setlength{\leftmargin}{.5in}%
\setlength{\rightmargin}{.5in}}
\item[\textbf{Rule 1:}] Two or more consecutive spaces in
text are the same as one.
\item[\textbf{Rule 2:}] A blank line (that is,
two end-of-line characters separated only by blanks and
tabs) indicates the end of a paragraph.
\end{list}
Rules 1 and~2 make typing and copying very convenient.
```

For further simple examples, you can look at various document class files to see how standard environments such as `verse`, `quote`, and so on, are defined.

Using counters

It is not very L^AT_EX-like to provide the numbers for the rules in the examples above. It would be more logical for L^AT_EX to do the numbering. The following is a more L^AT_EX-like coding of the second example:

```
\noindent Here are the most important \la\ rules about
spaces in text, sentences, and paragraphs:
\newcounter{spacerule}
```

```

\begin{list}{\textbf{Rule \arabic{spacerule}:}}
    {\setlength{\leftmargin}{.5in}
    \setlength{\rightmargin}{.5in}
    \usecounter{spacerule}}
\item Two or more consecutive spaces in text are the
same as one. \label{Li:Twoor}
\item A blank line (that is, two end-of-line
characters separated only by blanks and tabs)
indicates the end of a paragraph. \label{Li:blankline}
\end{list}
Rules \ref{Li:Twoor} and \ref{Li:blankline} make typing
and copying very convenient.

```

Note that

1. I declared the counter before the list environment with the line

```
\newcounter{spacerule}
```

2. I defined the *default_label* as

```
\textbf{Rule \arabic{spacerule}:}
```

3. In the *declarations*, I specified that the list should use the *spacerule* counter with the command

```
\usecounter{spacerule}
```

15.6.3 Two complete examples

In the previous examples, I set the values of `\leftmargin` and `\rightmargin`. The other length commands were not redefined, so their values remained the values set by the document class. In the following examples, I set the values of many more length commands.

Example 1 To get the following list:

┌

Here are the most important L^AT_EX rules about spaces in text, sentences, and paragraphs:

Rule 1: *Two or more consecutive spaces in text are the same as one.*

Rule 2: *A blank line—that is, two end-of-line characters separated only by blanks and tabs—indicates the end of a paragraph.*

Rules 1 and 2 make typing and copying very convenient.

we type

```
\noindent Here are the most important \la\ rules about
spaces in text, sentences, and paragraphs:
\newcounter{spacerule}
\begin{list}{\upshape\bfseries Rule \arabic{spacerule}:}
    {\setlength{\leftmargin}{1.5in}
    \setlength{\rightmargin}{0.6in}
    \setlength{\labelwidth}{1.0in}
    \setlength{\labelsep}{0.2in}
    \setlength{\parsep}{0.5ex plus 0.2ex
        minus 0.1ex}
    \setlength{\itemsep}{0ex plus 0.2ex
        minus 0ex}
    \usecounter{spacerule}
    \itshape}
    \item Two or more consecutive spaces in text are the
    same as one.\label{Li:Twoor}
    \item A blank line---that is, two end-of-line
    characters separated only by blanks and
        tabs---indicates
        the end of a paragraph.\label{Li:blankline}
\end{list}
Rules \ref{Li:Twoor} and~\ref{Li:blankline} make typing
and copying very convenient.
```

Note that

1. I declared the counter as in the previous example.
2. The last item in *declarations* is `\itshape`, which typesets the entire list in italics.
3. The *default_label* is defined as

```
\upshape\bfseries Rule \arabic{spacerule}
```

My first attempt was to define it as

```
\bfseries Rule \arabic{spacerule}
```

which typesets *myrule* in bold italics (because in Step 2 we set the whole list in italics). To force the label to be typeset upright, I start the *default_label* with the `\upshape` command.

4. The left margin is set to 1.5 inches and the right margin to 0.6 inches:

```
\setlength{\leftmargin}{1.5in}
\setlength{\rightmargin}{0.6in}
```

5. Next I set the width of the label to 1 inch, and the space between the label and the item to 0.2 inches:

```
\setlength{\labelwidth}{1.0in}
\setlength{\labelsep}{0.2in}
```

6. Finally, I set the paragraph separation to 0.5 ex, allowing stretching by 0.2 ex and shrinking by 0.1 ex, and the item separation to 0 ex, allowing stretching by 0.2 ex and no shrinking, by

```
\setlength{\parsep}{0.5ex plus 0.2ex minus 0.1ex}
\setlength{\itemsep}{0ex plus 0.2ex minus 0ex}
```

The actual amount of item separation is calculated by adding the values specified for `\parsep` and `\itemsep`.

A complicated list such as this should be defined as a new environment. For example, you could define a `myrules` environment:

```
\newenvironment{myrules}
{\begin{list}
{\upshape \bfseries Rule \arabic{spacerule}:}
{\setlength{\leftmargin}{1.5in}
\setlength{\rightmargin}{0.6in}
\setlength{\labelwidth}{1.0in}
\setlength{\labelsep}{0.2in}
\setlength{\parsep}{0.5ex plus 0.2ex minus 0.1ex}
\setlength{\itemsep}{0ex plus 0.2ex minus 0ex}
\usecounter{spacerule}
\itshape} }
{\end{list}}}
```

and then use it anywhere, as in

```
\begin{myrules}
\item Two or more consecutive spaces in text are the
same as one.\label{Li:Twoor}
\item A blank line---that is, two end-of-line
characters separated only by blanks and
tabs---indicates the end of a paragraph.
\label{Li:blankline}
Rules \ref{Li:Twoor} and~\ref{Li:blankline} make typing
and copying very convenient.
\end{myrules}
```

which typesets as the first example shown on page 455.

Example 2 In Section 3.6.2, we discussed the formatting of the following type of glossary:

┌	sentence a group of words terminated by a period, exclamation point, or question mark.
└	paragraph a group of sentences terminated by a blank line or by the new paragraph command.

Now we can create the glossary as a custom list:

```
\begin{list}{}
  {\setlength{\leftmargin}{30pt}
   \setlength{\rightmargin}{0pt}
   \setlength{\itemindent}{14pt}
   \setlength{\labelwidth}{40pt}
   \setlength{\labelsep}{5pt}
   \setlength{\parsep}{0.5ex plus 0.2ex minus 0.1ex}
   \setlength{\itemsep}{0ex plus 0.2ex minus 0ex}}
  \item[\textbf{sentence}\hfill] is a group of words
  terminated by a period, exclamation point,
  or question mark.
  \item[\textbf{paragraph}\hfill] is a group of sentences
  terminated by a blank line or by the \com{par} command.
\end{list}
```

There is nothing new in this example, except the `\hfill` commands in the optional arguments to left adjust the labels. With the long words in the example this adjustment is not necessary, but it would be needed for shorter words.

See Section 4.1 of LC3 on how to customize the three standard list environments and also for more complicated custom lists.

15.6.4 The *trivlist* environment

ℒ_{TEX} also provides a `trivlist` environment, of use for programmers. The environment is invoked in the form

```
\begin{trivlist}
  body
\end{trivlist}
```

It is similar to the `list` environment except that there are no arguments, and all the length commands are trivially set, most to 0 points, except for `\listparindent` and

`\parsep`, which are set to equal `\parindent` and `\parskip`, respectively. For instance, \LaTeX defines the `center` environment as follows:

```
\begin{trivlist}
  \centering \item[]
\end{trivlist}
```

15.7 The dangers of customization

We can customize \LaTeX in so many ways. We can add packages to expand its power and define new commands that better suit our work habits. These enhance \LaTeX and make it easier to work with. But they also introduce difficulties. Let us start with the obvious.

Whoever introduced the command `\textcompwordmark` knew that—even if we use command completion—we are not going to type

```
if\textcompwordmark f
```

to avoid having a ligature (see Section 3.4.6). It is a lot of typing, and the source file becomes hard to read. This cries out for a custom command, say, `\Iff`, which is short and *readable* (see Section 15.1.1).

When introducing custom commands, watch out for the following traps.

Trap 1 ■ Redefining a command that is a necessary part of \LaTeX .

This is easy to avoid. As discussed in Section 15.1.8, you can easily find out whether a command is already in use. If it is, do not redefine it unless you really know what you are doing.

Trap 2 ■ Defining too many commands.

This creates two problems. Your editor has a hard time making changes in your source file. And a few years later, when you want to reuse the material, you may have a difficult time understanding all those clever commands.

Trap 3 ■ Your contribution appears in a volume with many other authors and your custom commands create conflicts.

As your article appears in a publication, some parts of it are used for the whole volume. The title and maybe even the section titles are used in the table of contents.

The abstracts may be collected for the whole volume or there may be a joint bibliography.

There are now collections of thousands of math articles on the Internet. Write your articles so that even the editors of these collections can use them.



Rule: Do not use your own commands in the title of the article, in the abstract, in section titles, in the bibliography, or in captions of figures and tables.

Trap 4 ■ You submit the article to a journal that does not permit a separate custom command file.

For such journals, just copy the needed custom commands into the preamble of your article. Go through the list and **delete those custom commands that are not used in this article**. This helps the editor to look up your commands from a shorter list.



Rule: Introduce custom commands judiciously with very short names.

Introducing one-letter commands—for instance, using `\C` for the complex field—is dangerous because many one-letter commands are reserved by \LaTeX .

Two-letter custom commands are not quite this bad. Of the 2,500 or so possibilities only a few dozen are used by \LaTeX . The danger here is, of course, conflict with other authors and confusion for the editor. My command file has about 15 two-letter commands. For instance `\jj`, part of the `\jj`, `\JJ`, `\JJm` family. Also `\Id`, because Id is the standard notation for ideal lattices. Some editors may think that this is 15 too many.



Rule: Do not use `\def` to define your commands, with the exception of a very few delimited commands.

Using `\def` means giving up \LaTeX 's built in defenses. In the editorial office of my journal, about half the submitted articles that we cannot typeset violate this rule.



Rule: Do not redefine length commands, especially, if you do not know what other length commands are computed based on the ones you change.

The page layout diagram, Figure 10.4, should provide examples. Even simpler, *Do not redefine length commands*. Let the document class define them for your article.



Rule: Make sure that the packages you use are compatible.

For instance, the popular `psfrag` and `epsfig` packages cause problems if used with the AMS packages.

Be cautious when you use packages that redefine a lot of \LaTeX commands, such as `hyperref` (see Section 12.1).

You can read more about the plight of authors in the hands of incompetent editors in my article [22] and the difficult job of editors with articles violating the above rules in Enrico Gregorio [11].

PART VIII

Long Documents



BIB_TE_X

Oren Patashnik wrote the BIB_TE_X application (Version 0.98f in 1985, most recent version 0.99d in 2010) to assist L^AT_EX users in compiling bibliographies, especially long ones. Short bibliographies can easily be placed in the document directly (see Section 10.5.1).

It takes some effort to learn BIB_TE_X. But in the long run, the advantages of building bibliographic databases that can be reused and shared outweigh the disadvantage of a somewhat steep learning curve. The *bibliographic database files*, the `bib` files, contain the *bibliographic entries*. We discuss the format of these entries in Section 16.1, and then describe how to use BIB_TE_X to create bibliographies in Section 16.2.

BIB_TE_X uses a style, called a *bibliographic style*, or `bst` file, to format entries. On the next two pages we show the bibliography of the `art2bibtex.tex` sample article typeset with six different style files.

To simplify our discussion, in the rest of this chapter, I discuss only one style, the AMS plain style, `amspain.bst`, version 2.0. All of the examples shown are in this style, and several of the comments I make are true only for the AMS plain style. If you choose to use a different style, you should check its documentation for special rules.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-55281-6_16

- [1] Soo-Key Foo. *Lattice Constructions*. PhD thesis, University of Winnebago, Winnebago, MN, December 1990.
- [2] George A. Menuhin. *Universal Algebra*. D. van Nostrand, Princeton, 1968.
- [3] Ernest T. Moynahan. Ideals and congruence relations in lattices. II. *Magyar Tud. Akad. Mat. Fiz. Oszt. Közl.*, 7:417–434, 1957.
- [4] Ernest T. Moynahan. On a problem of M. Stone. *Acta Math. Acad. Sci. Hungar.*, 8:455–460, 1957.
- [5] Ferenc R. Richardson. *General Lattice Theory*. Mir, Moscow, expanded and revised edition, 1982.

plain.bst

- [Foo90] Soo-Key Foo. *Lattice Constructions*. PhD thesis, University of Winnebago, Winnebago, MN, December 1990.
- [Men68] George A. Menuhin. *Universal Algebra*. D. van Nostrand, Princeton, 1968.
- [Moy57a] Ernest T. Moynahan. Ideals and congruence relations in lattices. II. *Magyar Tud. Akad. Mat. Fiz. Oszt. Közl.*, 7:417–434, 1957.
- [Moy57b] Ernest T. Moynahan. On a problem of M. Stone. *Acta Math. Acad. Sci. Hungar.*, 8:455–460, 1957.
- [Ric82] Ferenc R. Richardson. *General Lattice Theory*. Mir, Moscow, expanded and revised edition, 1982.

alpha.bst

- 1. Soo-Key Foo, *Lattice constructions*, Ph.D. thesis, University of Winnebago, Winnebago, MN, December 1990.
- 2. George A. Menuhin, *Universal algebra*, D. van Nostrand, Princeton, 1968.
- 3. Ernest T. Moynahan, *Ideals and congruence relations in lattices*. II, Magyar Tud. Akad. Mat. Fiz. Oszt. Közl. **7** (1957), 417–434 (Hungarian).
- 4. ———, *On a problem of M. Stone*, Acta Math. Acad. Sci. Hungar. **8** (1957), 455–460.
- 5. Ferenc R. Richardson, *General lattice theory*, expanded and revised ed., Mir, Moscow, 1982 (Russian).

amsplain.bst

- [Foo90] Soo-Key Foo, *Lattice constructions*, Ph.D. thesis, University of Winnebago, Winnebago, MN, December 1990.
- [Men68] George A. Menuhin, *Universal algebra*, D. van Nostrand, Princeton, 1968.
- [Moy57a] Ernest T. Moynahan, *Ideals and congruence relations in lattices*. II, Magyar Tud. Akad. Mat. Fiz. Oszt. Közl. **7** (1957), 417–434 (Hungarian).
- [Moy57b] Ernest T. Moynahan, *On a problem of M. Stone*, Acta Math. Acad. Sci. Hungar. **8** (1957), 455–460.
- [Ric82] Ferenc R. Richardson, *General lattice theory*, expanded and revised ed., Mir, Moscow, 1982 (Russian).

amsalpha.bst

- [1] S.-K. Foo, *Lattice Constructions*, PhD thesis, University of Winnebago, Winnebago, MN, Dec. 1990.
- [2] G. A. Menuhin, *Universal Algebra*, D. van Nostrand, Princeton, 1968.
- [3] E. T. Moynahan, *Ideals and congruence relations in lattices. II*, Magyar Tud. Akad. Mat. Fiz. Oszt. Közl., 7 (1957), pp. 417–434.
- [4] ———, *On a problem of M. Stone*, Acta Math. Acad. Sci. Hungar., 8 (1957), pp. 455–460.
- [5] F. R. Richardson, *General Lattice Theory*, Mir, Moscow, expanded and revised ed., 1982.

siam.bst

- [1] F. R. Richardson, *General Lattice Theory*. Moscow: Mir, expanded and revised ed., 1982.
- [2] G. A. Menuhin, *Universal Algebra*. Princeton: D. van Nostrand, 1968.
- [3] E. T. Moynahan, “On a problem of M. Stone,” *Acta Math. Acad. Sci. Hungar.*, vol. 8, pp. 455–460, 1957.
- [4] S.-K. Foo, *Lattice Constructions*. PhD thesis, University of Winnebago, Winnebago, MN, Dec. 1990.
- [5] E. T. Moynahan, “Ideals and congruence relations in lattices. II,” *Magyar Tud. Akad. Mat. Fiz. Oszt. Közl.*, vol. 7, pp. 417–434, 1957.

ieeetr.bst

16.1 The database

A $\text{BIB}\text{T}\text{E}\text{X}$ database is a text file (using T1 or OT1 encoding; see Section 6.1) containing bibliographic entries in a strictly standardized format, which will be read and processed by the $\text{BIB}\text{T}\text{E}\text{X}$ application to produce a $\text{L}\text{T}\text{E}\text{X}$ bibliography file.

To use $\text{BIB}\text{T}\text{E}\text{X}$, you first have to learn how to assemble a database. This section explains how to do that.

There may be special tools available for your computer system that assist you in building and maintaining your bibliographic data. Such tools make compiling the data easier and may minimize formatting errors.

16.1.1 Entry types

A bibliographic entry is given in pieces called *fields*. The style (see Section 16.2.2) specifies how these fields are typeset. Here are two typical entries:

```
@BOOK{gm68,
  author = "George A. Menuhin",
  title = "Universal Algebra",
  publisher = "D.~Van Nostrand",
  address = "Princeton",
  year = 1968,
}
```

```
@ARTICLE{eM57,
  author = "Ernest T. Moynahan",
  title = "On a Problem of {M. Stone}",
  journal = "Acta Math. Acad. Sci. Hungar.",
  pages = "455-460",
  volume = 8,
  year = 1957,
}
```

The start of an entry is indicated with an at sign (@) followed by the *entry type*. In the first example, the entry type is BOOK, while in the second, it is ARTICLE. The entry type is followed by a left brace ({). The matching right brace (}) indicates the end of the entry. *BIB_TE_X* also allows you to use parentheses as delimiters for an entry. In this book, however, we use braces to enclose an entry.

The string @BOOK{ is followed by a *label*, gM68, which designates the name of the entry. Refer to this entry in your document with \cite{gM68}. The label is followed by a comma and a series of fields. In this example, there are five fields, *author*, *title*, *publisher*, *address*, and *year*. Each field starts with the field name, followed by = and the value of the field enclosed in double quotes ("). Be sure to use " and *not* *L_AT_EX* double quotes (‘ ‘ or ’ ’). Alternatively, *BIB_TE_X* also allows you to use braces to enclose the field value. In this book, we use double quotes to enclose a field.

Numeric field values, that is, fields consisting entirely of digits, do not need to be enclosed in double quotes or braces, for instance, *year* in the examples above, *volume* in the second example, and *number* in some of the examples that follow. Page ranges, such as 455-460, are not numeric field values since they contain -, so they must be enclosed in double quotes or braces.

There *must* be a comma before each field. The comma before the first field is placed after the label.

There are many standard entry types, including:

ARTICLE an article in a journal or magazine.

BOOK a book with an author (or editor) and a publisher.

BOOKLET a printed work without a publisher.

INBOOK a part of a book, such as a chapter or a page range that, in general, is not titled or authored separately.

INCOLLECTION a part of a book with its own title and perhaps author.

INPROCEEDINGS an article in a conference proceedings with its own title and author.

MANUAL technical documentation.

MASTERSTHESIS a master's thesis.

MISC an entry that does not fit in any other category.

PHDTHESIS a Ph.D. thesis.

PROCEEDINGS the proceedings of a conference.

TECHREPORT a report published by a school or institution.

UNPUBLISHED an unpublished paper.

Each entry includes a number of *fields* originally from the following list:

address	institution	pages
author	journal	publisher
booktitle	key	school
chapter	language	series
crossref	month	title
edition	note	type
editor	number	volume
howpublished	organization	year

You may also add fields for your own use. For example, you may want to add a `mycomments` field for personal comments.



Tip You may add as many fields as you want to. The style you choose determines which of the fields are actually used. All the others are ignored.

Commonly used examples of new field names include `annotate`, `Email`, `doi`, `URL`, `abstract`, `ISBN`, `keywords`, `mrnumber`, and so on. The `language` field is used by the AMS styles but not by any of the other styles mentioned in this chapter.



Tip `BIBTEX` does not care whether you use uppercase or lowercase letters (or mixed) for the names of entry types and fields. In this book, the entry types are shown in uppercase and field names in lowercase.



Tip Placing a comma after the last field is optional. I recommend that you put it there so that when you append a new field to the entry, the required comma separating the fields is present.

For each entry type there are both required and optional fields. Later in this section, I give two examples of each entry type. The first example of an entry type uses a small set of fields, while the second example is a larger one, showing more optional fields.

16.1.2 *Typing fields*

Make sure you type the field names correctly. If you misspell one, *BIBTEX* ignores the field. *BIBTEX* also warns you if a required field is missing. The author and editor fields require a name.



Rule: Names

1. Most names can be typed as usual, "Ernest T. Moynahan" or "Moynahan, Ernest T.", with one comma separating the family name from the given names.
2. Type two or more names separated by and. For instance,
author= "George Blue and Ernest Brown and Soo-Key Foo",
3. The family name of Miguel Lopez Fernandez is Lopez Fernandez, so type it as "Lopez Fernandez, Miguel". This informs *BIBTEX* that Lopez is not a middle name.
4. Type Orrin Frink, Jr. as "Frink, Jr., Orrin".

Rules 3 and 4 are seldom needed. In a bibliography of about 1,500 items, I found fewer than 10 names that could not be typed as usual. Note that you can type John von Neumann as "John von Neumann" or "von Neumann, John". Because *BIBTEX* knows about von, it handles the name properly.

There are a few rules concerning the title field.



Rule: Title

1. You should not put a period at the end of a title. The style supplies the appropriate punctuation.
2. Many styles, including the AMS styles, convert titles, except for the first letter of the title, to lowercase for all entry types. If you want a letter to appear in uppercase, put it—or the entire word—in braces. The same rule applies to the `edition` field. Some other styles only do this conversion for the titles of non-book-like entries.
3. To maximize the portability of your database, you should type titles with each important word capitalized:

```
title = "On a Problem of {M. Stone}",
```

The style used in this book, `amsplain.bst`, converts `Problem` to `problem`, so it makes no difference, but some styles do not. To be on the safe side, you should capitalize all words that may have to be capitalized.

For the record, here are the complete rules for titles:



Rule: Capitalize:

1. the first word;
 2. the first word in a subtitle (`BIBTEX` assumes that a subtitle follows a colon, so it capitalizes the first word after a colon—a colon not introducing a subtitle should be typed in braces);
 3. all other words except articles, unstressed conjunctions, and unstressed prepositions.
-

Words that should never be converted to lowercase, for example proper names such as Hilbert, should be enclosed in braces to prevent them from being converted to lowercase. In the example above, two letters in the title should not be converted to lowercase, so we enclosed `M. Stone` in braces. We could also have typed

```
{M. S}tone or {M.} {S}tone
```

`BIBTEX` and the style automatically handle a number of things for you that you would have to handle yourself when typing text.

1. You do not have to mark periods in abbreviations, as `._` in the names of journals (see Section 3.2.2). So

```
journal = "Acta Math. Acad. Sci. Hungar.",
```

typesets correctly.

2. You can type a single hyphen for a page range instead of the usual `--` in the `pages` field (see Section 3.4.2). So

```
pages = "455-460",
```

typesets correctly with an en dash.

3. You do not have to type nonbreakable spaces with `~` in the `author` or `editor` fields (see Section 3.4.3):

```
author = "George A. Menuhin",
```


is correct. Normally you would type George~A. Menuhin.

Finally, we state a rule about accented characters.



Rule: Accents

Put accented characters in braces: `{\{"a}}`.

This rule means that

```
author = "Paul Erd\H{o}s",
```

is not recommended. Instead, type

```
author = "Paul Erd{\H{o}}s",
```

This rule is, again, about portability. Some styles, e.g., alpha and amsalpha, create a citation for an article from the first three letters of the name and the last two digits of the year.

```
author = "Kurt G{\{"o}}del",
year = 1931,
```

creates the citation: [Göd31]. The accent is used only if the accents rule has been followed.

The downside of this rule is that the braces suppress kerning.

16.1.3 Articles

Entry type	ARTICLE
Required fields	author, title, journal, year, pages
Optional fields	volume, number, language, note

Examples:

- ┌

└

 1. Ernest T. Moynahan, *On a problem of M. Stone*, Acta Math. Acad. Sci. Hungar. **8** (1957), 455–460.
 2. Ernest T. Moynahan, *On a problem of M. Stone*, Acta Math. Acad. Sci. Hungar. **8** (1957), no. 5, 455–460 (English), Russian translation available.

typed as

```
@ARTICLE{eM57,
  author = "Ernest T. Moynahan",
  title = "On a Problem of {M. Stone}",
  journal = "Acta Math. Acad. Sci. Hungar.",
```

```

    pages = "455-460",
    volume = 8,
    year = 1957,
  }

@ARTICLE{eM57a,
  author = "Ernest T. Moynahan",
  title = "On a Problem of {M. Stone}",
  journal = "Acta Math. Acad. Sci. Hungar.",
  pages = "455-460",
  volume = 8,
  number = 5,
  year = 1957,
  note = "Russian translation available",
  language = "English",
}

```

16.1.4 Books

Entry type	BOOK
Required fields	author (or editor), title, publisher, year
Optional fields	edition, series, volume, number, address, month, language, note

Examples:

- ┌
1. George A. Menuhin, *Universal algebra*, D. Van Nostrand, Princeton, 1968.
 2. George A. Menuhin, *Universal algebra*, second ed., University Series in Higher Mathematics, vol. 58, D. Van Nostrand, Princeton, March 1968 (English), no Russian translation.
- └

typed as

```

@BOOK{gM68,
  author = "George A. Menuhin",
  title = "Universal Algebra",
  publisher = "D.~Van Nostrand",
  address = "Princeton",
  year = 1968,
}

```

```

@BOOK{gM68a,
  author = "George A. Menuhin",

```

```

title = "Universal Algebra",
publisher = "D.~Van Nostrand",
address = "Princeton",
year = 1968,
month = mar,
series = "University Series in Higher Mathematics",
volume = 58,
edition = "Second",
note = "no Russian translation",
language = "English",
}

```

Abbreviations, such as `mar`, are discussed in Section 16.1.9.

A second variant of `book` has an editor instead of an author:

```

┌
15. Robert S. Prescott (ed.), Universal algebra, D. Van Nostrand, Princeton,
└ 1968.
└

```

typed as

```

@BOOK{rP68,
  editor = "Robert S. Prescott",
  title = "Universal Algebra",
  publisher = "D.~Van Nostrand",
  address = "Princeton",
  year = 1968,
}

```

16.1.5 *Conference proceedings and collections*

Entry type	INPROCEEDINGS
Required fields	author, title, booktitle, year
Optional fields	address, editor, series, volume, number, organization, publisher, month, note, pages, language

Examples:

```

┌
7. Peter A. Konig, Composition of functions. Proceedings of the Conference on
  Universal Algebra, 1970.
8. Peter A. Konig, Composition of functions. Proceedings of the Conference on
  Universal Algebra (Kingston, ON) (G. H. Birnbaum, ed.), vol. 7, Cana-
  dian Mathematical Society, Queen's Univ., December 1970, available from
  the Montreal office, pp. 1–106 (English).
└

```

typed as

```
@INPROCEEDINGS{pK69,
  author = "Peter A. Konig",
  title = "Composition of Functions".
  booktitle = "Proceedings of the Conference on
    Universal Algebra",
  year = 1970,
}

@INPROCEEDINGS{pK69a,
  author = "Peter A. Konig",
  title = "Composition of Functions".
  booktitle = "Proceedings of the Conference on
    Universal Algebra",
  address = "Kingston, ON",
  publisher = "Queen's Univ.",
  organization = "Canadian Mathematical Society",
  editor = "G. H. Birnbaum",
  pages = "1-106",
  volume = 7,
  year = 1970,
  month = dec,
  language = "English",
}
```

The address field provides the location of the meeting. The address of the publisher should be in the publisher field and the address of the organization in the organization field.

Entry type	INCOLLECTION
Required fields	author, title, booktitle, publisher, year
Optional fields	editor, series, volume, number, address, edition, month, note, pages, language

Examples:

- ┌
1. Henry H. Albert, *Free torsoids*, Current Trends in Lattices, D. Van Nostrand, 1970.
 2. Henry H. Albert, *Free torsoids*, Current Trends in Lattices (George Burns, ed.), vol. 2, D. Van Nostrand, Princeton, January 1970, new edition is due next year, pp. 173–215 (German).
- └

is typed as

```
\noindent\verb+@INCOLLECTION{hA70,
```

```

author = "Henry H. Albert",
title = "Free Toroids",
booktitle = "Current Trends in Lattices".
publisher = "D.~Van Nostrand",
year = 1970,
}

```

```

@INCOLLECTION{hA70a,
  author = "Henry H. Albert",
  editor = "George Burns",
  title = "Free Toroids",
  booktitle = "Current Trends in Lattices".
  publisher = "D.~Van Nostrand",
  address = "Princeton",
  pages = "173-215",
  volume = 2,
  year = 1970,
  month = jan,
  note = "new edition is due next year",
  language = "German",
}

```

The address field contains the address of the publisher.

Cross-referencing

If your database has several articles from the same conference proceedings and collections, you may prefer to make an entry for the entire volume, and cross-reference individual articles to that entry. For instance,

```

@PROCEEDINGS{UA69,
  title = "Proceedings of the Conference on,
    Universal Algebra",
  booktitle = "Proceedings of the Conference on
    Universal Algebra",
  address = "Kingston, ON",
  publisher = "Canadian Mathematical Society",
  editor = "G. H. Birnbaum",
  volume = 7,
  year = 1970,
}

```

may be the entry for the proceedings volume as a whole, and

```

@INPROCEEDINGS{pK69a,

```

```

author = "Peter A. Konig",
title = "Composition of Functions",
booktitle = "Proceedings of the Conference on
    Universal Algebra",
pages = "1-106",
crossref = "UA69",
}

```

is the cross-referencing entry for a specific article. These two entries produce the following:

- ┌
1. G. H. Birnbaum (ed.), *Proceedings of the conference on universal algebra*, vol. 7, Kingston, ON, Canadian Mathematical Society, 1970.
 2. Peter A. Konig, *Composition of functions*, in Birnbaum [1], pp. 1–106.
- └



Rule: Cross-references

1. All the required fields of the cross-referencing entry must appear in either that entry or in the cross-referenced entry.
 2. The cross-referenced entry should have both a `title` and a `booktitle` field.
 3. The cross-referenced entry must appear in the `bib` file later than any entry that cross-references it.
-

16.1.6 Theses

Entry type	MASTERSTHESIS or PHDTHESIS
Required fields	author, title, school, year
Optional fields	type, address, month, note, pages

Examples:

- ┌
1. Soo-Key Foo, *Lattice constructions*, Ph.D. thesis, University of Winnebago, 1990.
 2. Soo-Key Foo, *Lattice constructions*, Ph.D. dissertation, University of Winnebago, Winnebago, MN, December 1990, final revision not yet available, pp. 1–126.
- └

is typed as

```

@PHDTHESIS{sF90,
  author = "Soo-Key Foo",
  title = "Lattice Constructions",
  school = "University of Winnebago",
  year = 1990,
}

@PHDTHESIS{sF90a,
  author = "Soo-Key Foo",
  title = "Lattice Constructions",
  school = "University of Winnebago",
  address = "Winnebago, MN",
  year = 1990,
  month = dec,
  note = "final revision not yet available",
  type = "Ph.D. dissertation",
  pages = "1-126",
}

```

If the `type` field is present, its content takes the place of the phrase Ph.D. thesis (or Master's thesis).

16.1.7 *Technical reports*

Entry type	TECHREPORT
Required fields	author, title, institution, year
Optional fields	type, number, address, month, note

Examples:

- ┌
1. Grant H. Foster, *Computational complexity in lattice theory*, tech. report, Carnegie Mellon University, 1986.
 2. Grant H. Foster, *Computational complexity in lattice theory*, Research Note 128A, Carnegie Mellon University, Pittsburgh, PA, December 1986, in preparation.
- └

is typed as

```

@TECHREPORT{gF86,
  author = "Grant H. Foster",
  title = "Computational Complexity in Lattice Theory",
  institution = "Carnegie Mellon University",
  year = 1986,
}

```

```
@TECHREPORT{gF86a,
  author = "Grant H. Foster",
  title = "Computational Complexity in Lattice Theory",
  institution = "Carnegie Mellon University",
  year = 1986,
  month = dec,
  type = "Research Note",
  address = "Pittsburgh, PA",
  number = "128A",
  note = "in preparation",
}
```

16.1.8 Manuscripts and other entry types

Entry type	UNPUBLISHED
Required fields	author, title, note
Optional fields	month, year

Examples:

- ┌
1. William A. Landau, *Representations of complete lattices*, manuscript, 55 pages.
 2. William A. Landau, *Representations of complete lattices*, manuscript, 55 pages, December 1975.
- └

is typed as

```
@UNPUBLISHED{wL75,
  author = "William A. Landau",
  title = "Representations of Complete Lattices",
  note = "manuscript, 55~pages",
}
```

```
@UNPUBLISHED{wL75a,
  author = "William A. Landau",
  title = "Representations of Complete Lattices",
  year = 1975,
  month = dec,
  note = "manuscript, 55~pages",
}
```

Other standard entry types include

Entry type	BOOKLET
Required field	title
Optional fields	author, howpublished, address, month, year, note
Entry type	INBOOK
Required fields	author or editor, title, chapter or pages, publisher, year
Optional fields	series, volume, number, type, address, edition, month, pages, language, note
Entry type	MANUAL
Required field	title
Optional fields	author, organization, address, edition, month, year, note
Entry type	MISC
Required field	at least one of the optional fields must be present
Optional fields	author, title, howpublished, month, year, note, pages
Entry type	PROCEEDINGS
Required fields	title, year
Optional fields	editor, series, volume, number, address, organization, publisher, month, note

16.1.9 Abbreviations

You may have noticed the field `month = dec` in some of the examples. This field uses an abbreviation. Most *BIBTEX* styles, including the AMS styles, include abbreviations for the months of the year: `jan`, `feb`, ..., `dec`. When an abbreviation is used, it is not enclosed in quotes (") or braces ({ }). The style defines what is actually to be typeset. Most styles typeset `dec` as either `Dec.` or `December`.

The name of the abbreviation, such as `dec`, is a string of characters that starts with a letter, does not contain a space, an equal sign (=), a comma, or any of the special characters listed in Section 3.4.4.

You may define your own abbreviations using the command `@STRING`. For example,

```
@STRING{au = "Algebra Universalis"}
```

A string definition can be placed anywhere in a `bib` file, as long as it precedes the first use of the abbreviation in an entry.

The AMS supplies the `mrabbrev.bib` file containing the standard abbreviations for many mathematical journals. Based on this file, you can make your own `abbrev.bib` file containing entries for all the journals you reference with whatever abbreviations you find easiest to remember.

If you use this scheme, the command you use to specify the `bib` files may look like

```
\bibliography{abbrev,... }
```

Section 16.2.1 explains the `\bibliography` command.

16.2 Using BIB_TE_X

In Section 16.1, you learned how to create database files. The sample bib file is `art2bibtex.bib` in the `samples` folder (see page 5). In this section, you learn how to use BIB_TE_X to process these files to create a bibliography. We illustrate the process of working with BIB_TE_X with the `art2b` sample article.

We use the `amsplain` style. To obtain all six examples of different styles shown on pages 466–467, just change `amsplain` to the appropriate style name in your document and typeset it.

One BIB_TE_X style behaves differently. The `apacite` style of the American Psychological Association requires that the preamble of your document include the line

```
\usepackage{apalike}
```

in addition to using the style file. The package can also be modified by a large number of options.

16.2.1 Sample files

Type the following two lines to replace the `thebibliography` environment in the `art2.tex` sample document:

```
\bibliographystyle{amsplain}
\bibliography{bibtextempl}
```

Save the new sample article as `art2bibtex.tex`. The first line specifies the bst file, `amsplain.bst`, which is part of the AMS distribution (see Section 11.6). The second line specifies the database files used.

The contents of the `bibtextempl.bib` bibliographic database file are as follows:

```
@BOOK{gM68,
  author = "George A. Menuhin",
  title = "Universal Algebra",
  publisher = "D.~Van Nostrand",
  address = "Princeton",
  year = 1968,
}
@BOOK{fR82,
  author = "Ferenc R. Richardson",
  title = "General Lattice Theory",
  edition = "Expanded and Revised",
```

```

    language = "Russian",
    publisher = "Mir",
    address = "Moscow",
    year = 1982,
  }
@ARTICLE{eM57,
  author = "Ernest T. Moynahan",
  title = "On a Problem of {M. Stone}",
  journal = "Acta Math. Acad. Sci. Hungar.",
  pages = "455-460",
  volume = 8,
  year = 1957,
}

@ARTICLE{eM57a,
  author = "Ernest T. Moynahan",
  title = "Ideals and Congruence Relations in
    Lattices.~\textup{II}",
  journal = "Magyar Tud. Akad. Mat. Fiz. Oszt. K{\o}zl.",
  language = "Hungarian",
  pages = "417-434",
  volume = 7,
  year = 1957,
}

@PHDTHESIS{sF90,
  author = "Soo-Key Foo",
  title = "Lattice Constructions",
  school = "University of Winnebago",
  address = "Winnebago, MN",
  year = 1990,
  month = dec,
}

```

16.2.2 *Setup*

Before you start *BIB_TE_X*, make sure that everything is set up properly as described in this section.

To list database entries in the bibliography, use the `\cite` command. Refer to Section 10.5.1 for details on how to use citations. If you want to have a reference listed in the bibliography without a citation in the text, then use the `\nocite` command. For example,

```
\cite{pK57}
```

includes the reference in the bibliography and cites the entry with label pK57, whereas

```
\nocite{pK57}
```

includes the reference in the bibliography but does not cite the entry. In either case, one of the bib files specified in the argument of the `\bibliography` command must contain an entry with the label pK57. The `\nocite{*}` command includes *all* the entries from the bibliographic databases you’ve specified.

Your document must specify the bibliography style and must name the bib files to be used. For instance, the `art2bibtex.tex` sample article contains the lines

```
\bibliographystyle{amsplain}
\bibliography{art2b}
```

The `\bibliographystyle` command specifies `amsplain.bst` as the style and the `\bibliography` command specifies the database file `art2bibtex.bib`. To use several database files, separate them with commas, as in

```
\bibliography{abbrev,gg,lattice,art2b}
```

where:

- `abbrev.bib` contains custom abbreviations;
- `gg.bib` contains personal articles;
- `lattice.bib` contains lattice theory articles by other authors;
- `art2bibtex.bib` contains additional references needed.

It is important to make sure that the `bst` file, the `bib` file(s), and the \LaTeX document(s) are in folders where *BIBTEX* can find them. If you are just starting out, you can simply copy all of them into one folder. Later, you may want to look for a more permanent solution by keeping the files `abbrev.bib` and `lattice.bib` in one “central” location, while placing `art2bibtex.bib` in the same folder as its corresponding \LaTeX document.

16.2.3 *Four steps of BIBTEXing*

The following steps produce a typeset bibliography in your \LaTeX document. We use the `art2bibtex.tex` sample article as an example. In the figure, `sampartb.tex` stands for `art2bibtex.tex`.

Step 1 Check that *BIBTEX*, your \LaTeX document, and the bib files are placed in the appropriate folders.

Step 2 Typeset `art2bibtex.tex` to get a fresh aux file. This step is illustrated in Figure 16.1.

Step 3 Run *BIB_TE_X* on the `art2bibtex.aux` file in one of the following three ways:

- by invoking it with the argument `art2b`;
- by starting the application and then opening `art2bibtex.aux`;
- by running it by choosing it as a menu option of your editor or GUI front end or by clicking on an icon.

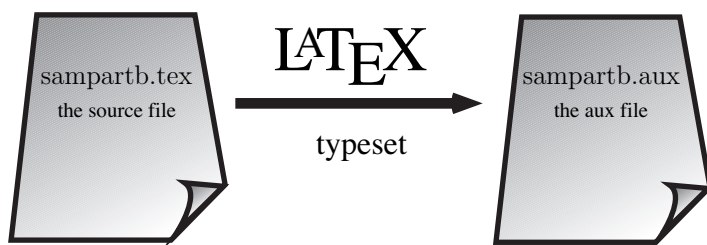


Figure 16.1: Using *BIB_TE_X*, step 2.

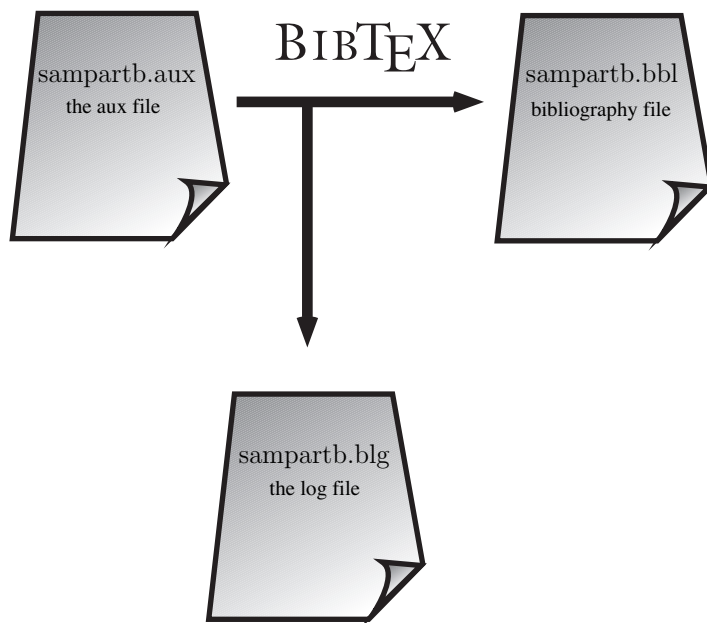


Figure 16.2: Using *BIB_TE_X*, step 3.

If *BIBTEX* cannot find a crucial file, for example, the `bst` file, it stops. The reason it stopped is shown in the log window and also written to a `blg` (bibliography log) file, `art2bibtex.blg`. Correct the error(s) and go back to step 2. A successful run creates a `bb1` (bibliography) file, `art2bibtex.bb1`, in addition to `art2bibtex.blg`. This step is illustrated in Figure 16.2.

Step 4 Typeset the *L*^AT_EX document `art2bibtex.tex` *twice*.

16.2.4 *BIBTEX* files

BIBTEX uses and creates a number of files when it is run. To illustrate this process, complete the four steps using `art2bibtex.tex`.

Step 1 Start fresh by deleting the `aux`, `blg`, and `bb1` files, if they are present.

Step 2 Typeset the article `art2bibtex.tex` to get an `aux` file (see Figure 16.1). Notice that the log file contains warnings about missing references and a number of other lines not relevant to the current discussion. The lines in the `aux` file containing bibliographic information are

```
\citation{fR82}
\citation{gM68}
\citation{eM57}
\citation{sF90}
\citation{eM57a}
\bibstyle{amsplain}
\bibdata{art2b}
```

Each `\citation` command in this file corresponds to a `\cite` or `\nocite` command in the article. The lines

```
\bibliographystyle{amsplain}
\bibliography{art2b}
```

in `art2bibtex.tex` are written as

```
\bibstyle{amsplain}
\bibdata{art2b}
```

in the `art2bibtex.aux` file.

Step 3 As illustrated in Figure 16.2, run *BIBTEX* on the `art2bibtex.aux` file (really, on the `art2bibtex.tex` file). How we do this, depends on the *L*^AT_EX installation you have.

BIBTEX generates two new files: `art2bibtex.blg` and `art2bibtex.bb1`. Look at `art2bibtex.blg`:

This is BibTeX, C Version 0.99c
 The top-level auxiliary file: art2bibtex.aux
 The style: amsplain.bst
 Database file #1: art2bibtex.bib

On some systems, this file may be much longer than the one I show here. At present, this blg file does not contain much important information. If there were any warnings or errors, they would be listed in this file.

The art2bibtex.bbl file, in which BIB_TE_X created a thebibliography environment (see Section 10.5.1) is more interesting:

```
\providecommand{\bysame}{\leavevmode
\hbox to3em {\hrulefill}\thinspace}
\begin{thebibliography}{1}

\bibitem{sF90}
Soo-Key Foo, \emph{Lattice constructions},
Ph.D. thesis, University of Winnebago,
Winnebago, MN, December 1990.

\bibitem{gM68}
George~A. Menuhin, \emph{Universal algebra},
D.~Van Nostrand, Princeton, 1968.

\bibitem{eM57a}
Ernest~T. Moynahan, \emph{Ideals and congruence
relations in lattices.~\textup{II}},
Magyar Tud. Akad. Mat. Fiz. Oszt. K{\"}{o}zl.
\textbf{7} (1957), 417-434 (Hungarian).

\bibitem{eM57}
\bysame, \emph{On a problem of {M. Stone}}, Acta
Math. Acad. Sci. Hungar. \textbf{8} (1957),
455-460.

\bibitem{fR82}
Ferenc~R. Richardson, \emph{General lattice theory},
expanded and revised ed., Mir, Moscow,
1982 (Russian).

\end{thebibliography}
```

Observe that the nonbreakable spaces (ties) and the \bysame command have

been provided in the author fields.

Step 4 Now typeset `art2bibtex.tex` again. The typeset version now has a REFERENCES section, constructed from the `bb1` file, but the new `log` file has warnings about missing entries. The new `aux` file contains five interesting new lines:

```
\bibcite{sF90}{1}
\bibcite{gM68}{2}
\bibcite{eM57a}{3}
\bibcite{eM57}{4}
\bibcite{fr82}{5}
```

These lines identify the cross-reference label `sF90` (see the first line shown—the symbol designates Foo’s thesis in `art2bibtex.bib`) with the number 1, and so on. Now typeset `art2bibtex.tex` again, and all the citations are correctly placed in the typeset article.

Observe:

1. The crucial step 3, running the *BIB_TE_X* application, gives different messages and obeys different rules from *L^AT_EX*—see Section 16.2.5.
2. The `art2bibtex.bb1` file was created by *BIB_TE_X*. It is not changed by running *L^AT_EX*.

16.2.5 *BIB_TE_X rules and messages*



Rule: *BIB_TE_X* and %

You cannot comment out a field with %.

For example, the entry

```
@ARTICLE{eM57,
  author = "Ernest T. Moynahan",
  title = "On a Problem of {M. Stone}",
  journal = "Acta Math. Acad. Sci. Hungar.",
  % pages = "455-460",
  volume = 8,
  year = 1957,
}
```

causes *BIB_TE_X* to generate the message:

You’re missing a field name

Recall that *BIBTEX* ignores field names it cannot recognize. So changing the field name pages, for example to pages-comment, does not give an error message. However, doing so removes a required field, so you get the warning message

Warning--missing pages in eM57



Rule: *BIBTEX* field names

Do not abbreviate field names.

For instance, if you abbreviate volume to vol, as in

```
@ARTICLE{eM57,
  author = "Ernest T. Moynahan",
  title = "On a Problem of {M. Stone}",
  journal = "Acta Math. Acad. Sci. Hungar.",
  pages = "455-460",
  vol = 8,
  year = 1957,
}
```

the vol field is simply ignored. This entry is typeset as

- ┌
 3. Ernest T. Moynahan, *On a problem of M. Stone*, Acta Math. Acad. Sci.
 Hungar. (1957), 455–460.
 └

instead of

- ┌
 3. Ernest T. Moynahan, *On a problem of M. Stone*, Acta Math. Acad. Sci.
 Hungar. **8** (1957), 455–460.
 └



Rule: *BIBTEX* field terminations

Make sure that every field of an entry, except possibly the last, is terminated with a comma.

If you drop a comma before a field, you get a message such as

```
I was expecting a ',' or a ')'
                line 6 of file art2bibtex.bib
:
:   year = 1968,
(may have been on previous line)
```

I'm skipping whatever remains of this entry
Warning--missing year in gM68+



Rule: $\text{BIB}\text{T}\text{E}\text{X}$ field value terminations

Make sure that the field value is properly terminated.

You should be careful not to drop a double quote or brace. If you drop the closing quote on line 11 of the bib file,

```
title = "General Lattice Theory
```

you get the message

```
I was expecting a ',' or a '}'
                        line 12 of file art2bibtex.bib
:   edition = "
:           Expanded and Revised",
I'm skipping whatever remains of this entry
Warning--missing publisher in fR82
Warning--missing year in fR82
```

If, instead, you drop the opening double quote in the same line, you get the message

```
Warning--string name "general" is undefined
--line 11 of file art2bibtex.bib
```

$\text{BIB}\text{T}\text{E}\text{X}$ assumed that `general` was an abbreviation, since it was not preceded by a ".".

The obvious conclusion is that you have to be very careful about typing your bibliographic entries for $\text{BIB}\text{T}\text{E}\text{X}$.

16.2.6 *Submitting an article*

If you submit an article to a journal that provides you with a $\text{BIB}\text{T}\text{E}\text{X}$ style file, then you can submit the article and the $\text{BIB}\text{T}\text{E}\text{X}$ database file, pared down of course. If this is not the case, create the `bbl` file with `amsplain.bst` and copy and paste the content into the `thebibliography` environment in the article. Then the journal's editor can edit the bibliography.

16.2.7 *Some resources*

There is a lot more to $\text{BIB}\text{T}\text{E}\text{X}$ than what has been covered in this chapter. For example, $\text{BIB}\text{T}\text{E}\text{X}$'s algorithm to alphabetize names is fairly complicated. Some names create additional difficulties. Where should John von Neumann be placed, under the "v"-s or

the “N”-s? It depends on the style. How do we handle names where the first word is the family name, as in Ho Chi Minh or Grätzer György? Again, it depends on the style.

Oren Patashnik [36] has many helpful hints. It includes a clever hack to order entries correctly even when the style does not do so. Chapter 15 of LC3 has a long discussion of BIB_TE_X. It also contains a long list of style files in five tables: 15.7–15.10.

You can easily build your own mathematical databases with MathSciNet from the AMS . Do a search. When the result page comes up, select the articles you want (or click on **Select All** and then click on **Export**. For the format, select **BibTeX** and click on the rectangle **Copy**. The references are pasted onto the clipboard, so you can copy them into your BIB_TE_X file.



Beyond BIBTEX

17.1 BibL^AT_EX

17.1.1 Introduction

The AMS released `amsrefs`, the kid brother of BIBTEX, at its annual meeting in January 2002. The presentation was made by Michael Downes, who designed and coded the package. It was a pleasure to behold. I wrote up a chapter on it for the next edition of my *Math into L^AT_EX* book. 30 pages described `amsrefs`. To define, say, the article format, a few lines of short L^AT_EX commands would do. After Michael Downes passed away, the AMS took the project in a different direction.

17.1.2 BibL^AT_EX overview

BibL^AT_EX is a package for L^AT_EX, designed for formatting and managing bibliographic references and citations. It's a successor to the traditional BIBTEX system, discussed in the previous chapter.

Here are some features of BibL^AT_EX:

Customizable Formatting BibL^AT_EX allows users to customize the formatting of their

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-55281-6_17

bibliography and citations extensively without having to modify BibTeX style files. Instead, formats are defined within LaTeX commands.

Multiple Bibliographies It is easy to generate multiple bibliographies in a single document, such as separate lists of publications, online sources, and primary sources.

Unicode Support BibLaTeX supports Unicode, which is crucial for correctly rendering bibliographies in languages that use non-Latin scripts.

Localization It provides localization for many languages, meaning it can automatically adapt the bibliography and citation format to different linguistic conventions.

Complex Data Models BibLaTeX can handle more complex data models than traditional BibTeX, allowing for more nuanced citation types and relationships.

In this book, we focus on citations.

17.1.3 *biber*

biber is a “backend” for BibLaTeX. The term “backend” refers to the application responsible for handling the data and operations related to the bibliography. biber is one such backend, known for its capabilities.

Our sample file is `art2biblatex`, which starts out as `art2bibtex`. We invoke the BibLaTeX package and specify biber as the backend by adding the line:

```
\usepackage[backend=biber]{biblatex}
```

We also add the line

```
\addbibresource{bibtexlong.bib}
```

to specify the bibliography database file to be used.

You use BibLaTeX the way you use BibTeX; see Section 16.2.3, with biber replacing BibTeX.

17.1.4 Citations

BibLaTeX provides several citation commands to fit different contexts and styles within your document. Figure 17.1 lists a few. For the complete list; see Philipp Lehman [33].

Figure 17.1 is the typeset `art3.tex` (see sample files):

```
% Sample file: art3.tex
\documentclass{amsart}
\usepackage{amssymb,latexsym}
\usepackage[backend=biber]{biblatex}
\addbibresource{bibtexlong.bib}
```

Bib \LaTeX CITATION COMMANDS

Bib \LaTeX has very many citation commands. We list some.

- Command `\cite{eM57}` produces a basic citation.
Typesets as [1]
- Command `\parencite{eM57}` produces a parenthetical citation.
Typesets as [1]
- Command `\footcite{eM57}` produces a footnote citation.
Typesets as¹
- Command `\textcite{eM57}` produces an in-text citations.
Typesets as Moynahan [1]
- Command `\citeauthor{eM57}` cites just the author(s).
Typesets as Moynahan
- Command `\citetitle{eM57}` cites just the title of the entry.
Typesets as “On a Problem of M. Stone”
- Command `\citeyear{eM57}` cites only the year of the entry..
Typesets as 1957
- Command `\fullcite{eM57}` provides the full reference.
Typesets as Ernest T. Moynahan. “On a Problem of M. Stone”. In:
Acta Math. Acad. Sci. Hungar. 8 (1957), pp. 455–460

For multiple citations at once, use `\cites` for citations such as [1, 3, 5]),
`\parencites`, `\footcites`, and `\textcites`.

REFERENCES

- [1] Ernest T. Moynahan, *On a problem of M. Stone*, *Acta Math. Acad. Sci. Hungar.* **8** (1957), 455–460.

Date: March 15, 2024.

¹1.

Figure 17.1: Some Bib \LaTeX citation commands

```

\usepackage{xspace}
\newcommand{\biblatex}{Bib\hspace{.05em}\latex}
\newcommand{\latex}{\textup{\LaTeX}\xspace}

\begin{document}
\title{\biblatex citation commands}
\date{March 15, 2024}
\maketitle

\biblatex\index{biblatex@\protect\biblatex}
has very many citation commands.
We list some.
\begin{itemize}
\item Command \verb+\cite{eM57}+ produces a basic citation.

Typesets as \cite{eM57}
\item Command \verb+\parencite{eM57}+ produces
a parenthetical citation.

Typesets as \parencite{eM57}
\item Command \verb+\footcite{eM57}+ produces
a footnote citation.

Typesets as \footcite{eM57}
\item Command \verb+\textcite{eM57}+ produces
an in-text citations.

Typesets as \textcite{eM57}
\item Command \verb+\citeauthor{eM57}+ cites just the author(s).

Typesets as \citeauthor{eM57}
\item Command \verb+\citetitle{eM57}+ cites
just the title of the entry.

Typesets as \citetitle{eM57}
\item Command \verb+\citeyear{eM57}+ cites
only the year of the entry.

Typesets as \citeyear{eM57}
\item Command \verb+\fullcite{eM57}+ provides the full reference.

Typesets as \fullcite{eM57}

```

```
\end{itemize}
```

For multiple citations at once, use `\verb+\cites+` for citations such as [1, 3, 5]), `\verb+\parencites+`, `\verb+\footcites+`, and `\verb+\textcites+`.

```
\begin{thebibliography}{9}
  \bibitem{eM57}
    Ernest~T. Moynahan, \emph{On a problem of M. Stone},
    Acta Math. Acad. Sci. Hungar. \textbf{8} (1957), 455--460.
\end{thebibliography}
\end{document}
```

17.1.5 *My citations*

Curiously, the citation form most often used in mathematics (H. Lakser [23]), is not standard in Bib \LaTeX . For instance, LC3 dedicates over a 110 pages to citations, but this citation form never comes up—showing the interesting split between mathematics and computer science.

I show how to introduce the `\mystyle` that displays H. Lakser [23]. Here is the sample file `art4.tex`:

```
% Sample file: art4.tex
\documentclass{amsart}
\usepackage[style=numeric, backend=biber]{biblatex}
\addbibresource{mybibliography.bib}

\DeclareCiteCommand{\mycite}
  {\usebibmacro{prenote}}
  {\ifciteindex
    {\indexnames{labelname}}
    {}%
    \printnames[first-last,firstinits=true,
      terseinits=true]{labelname}%
    \addcomma\space
    \printfield{prefixnumber}%
    \printfield{labelnumber}}
  {\multicitedelim}
  {\usebibmacro{postnote}}

\begin{document}
\mycite{key}
\printbibliography
```



```
\end{document}
```

Now follow the four steps in Section 16.2.3, except that in Step 3 run `biber`. Now the typeset file will display: H.Lakser [1].

How to run `biber` depends on the operating system and the T_EX distribution.

17.1.6 Cross-referencing between entries

To implement cross-referencing between entries, use the `crossref` field, as in the sample file `bibtexcross.bib`:

```
@BOOK(uabook,
  author = "George~A. Menuhin",
  title = "Universal Algebra",
  publisher = "D.~Van Nostrand",
  address = "Princeton",
  year = 1968,
)

@INCOLLECTION(Albertchapter,
  author = "Henry~H. Albert",
  title = "Free Torsoids",
  crossref = {uabook},
  publisher = "D.~Van Nostrand",
  year = 1970,
)
```

When citing `mychapter`, Bib_{La}T_EX will include the information from `uabook`.

17.1.7 Sorting

You can use `biber` to sort the bib file. The most common sorting options are:

ynt Sort by year, name, title.

net Sort by name, editor, title.

nty Sort by name, title, year.

anyt Sort by author, name, year, title.

`biber` is a command line tool, ask ChatGPT how to use it in your operating system.

17.2 Research tools

Over the years, several companion tools and resources have been developed to enhance $\text{BIB}\text{T}_{\text{E}}\text{X}$'s functionality. Here's a list of some prominent ones:

JabRef An open-source bibliography reference manager. It provides a graphical interface to manage $\text{BIB}\text{T}_{\text{E}}\text{X}$ databases, making it easier to organize and search references.

Mendeley A reference manager and academic social network that can help researchers manage their references and produce $\text{BIB}\text{T}_{\text{E}}\text{X}$ files for $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ documents.

Zotero Another reference manager that supports collecting, organizing, citing, and sharing research. It can export collections in the $\text{BIB}\text{T}_{\text{E}}\text{X}$ format.

BibTool A command-line tool that offers various operations on $\text{BIB}\text{T}_{\text{E}}\text{X}$ databases, such as formatting, sorting, and other manipulations.

Citeulike A free online service to manage and discover scholarly references. It can export references in $\text{BIB}\text{T}_{\text{E}}\text{X}$ format.

Google Scholar While primarily known as an academic search engine, it allows users to export citations in $\text{BIB}\text{T}_{\text{E}}\text{X}$ format.

Bibtex2html A tool that translates $\text{BIB}\text{T}_{\text{E}}\text{X}$ bibliographies to HTML format, allowing for easy web publication of bibliographies.

CrossTeX An evolution of $\text{BIB}\text{T}_{\text{E}}\text{X}$ that aims to modernize and enrich the types and quality of citations. It offers a more modern database format and better cross-referencing capabilities.

Bibsonomy A social bookmarking and publication-sharing system. It allows users to bookmark and share their publication lists and offers $\text{BIB}\text{T}_{\text{E}}\text{X}$ export functionality.

For instance, to sort a `bib` file, use `BibTool`, `JabRef`, or `biber`. For a straightforward task of sorting a `bib` file, `BibTool` might be the most direct and efficient option if you're comfortable with command-line tools. For a graphical interface, `JabRef` would be the choice.

As another example, you have a PDF file and you wish to extract the bibliographic information and store it in a $\text{BIB}\text{T}_{\text{E}}\text{X}$ file and attach the PDF file to the $\text{BIB}\text{T}_{\text{E}}\text{X}$ file permanently, so it is available when needed. Here's how you can do it:

PDFtoText Extract text from PDF. This tool that's a part of the `Xpdf` package. It converts a PDF to plain text.

AnyStyle Extract bibliographic Information. It parses references from plain text. After extracting the text from the PDF, you can feed the bibliography section to AnyStyle, and it will parse the references and can output in B_IB_TE_X format.

CERMINE An extraction tool to process articles in PDF format and extract metadata and referenced bibliographic items. It can output the results in B_IB_TE_X format.

Of course, you use JabRef to organize, manage, and store the bibliographic information.



MakeIndex

Pehong Chen's *MakeIndex* application, described in Pehong Chen and Michael A. Harrison's *Index preparation and processing* [5], helps L^AT_EX users create long indexes. For a short index, you can easily do without it (see Section 10.5.2).

In Section 18.1, we show you by an example how to prepare an article for indexing. We introduce formally the index commands in Section 18.2. In Section 18.3, we describe how L^AT_EX and *MakeIndex* process the index entries. The rules are stated in Section 18.4.

Multiple indexes are almost as easy as single indexes. They are described in Section 18.5. We conclude with glossaries in Section 18.6.

Indexing is a difficult task. For an extensive discussion on how to create a useful index, consult *The Chicago Manual of Style*, 15th edition [6].

18.1 Preparing the document

L^AT_EX provides the `theindex` environment (see Section 10.5.2). Within this environment, it provides the `\item`, `\subitem`, and `\subsubitem` commands to typeset entries, subentries, and subsubentries, respectively, and the `\indexspace` command for adding vertical space between alphabetical blocks; see Figure 18.1 for an example.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-55281-6_18

The `makeidx` package provides the `\index` command for specifying the index entry at a particular point in the document, which becomes a page reference for the entry in the typeset index.

Making an index entry with *MakeIndex* is easy. You simply place the index commands in your source file, and then let \LaTeX and *MakeIndex* do the work of gathering the entries and the page numbers for the entries, sorting them, and formatting the typeset index.

There are three steps:

1. In the preamble of your \LaTeX document, include the line

```
\makeindex
```

If you do not use an AMS document class, include the two lines

```
\usepackage{makeidx}  
\makeindex
```

2. Type the line

```
\printindex
```

at the point in your document where you want the index to appear, usually as part of the back matter (see Section 10.5).

3. Mark all entries in your document with `\index` commands.

We illustrate this procedure with the `art1ind.tex` article, which modifies the article `art1.tex` by inserting a number of index entries (both these files are in the `samples` folder; see page 5).

We now add a dozen `\index` commands to `art1.tex`.

Command 1

Retype the line

```
\begin{theorem}
```

to read

```
\begin{theorem}\index{Main Theorem}
```

Commands 2 and 3

Type the commands

```
\index{pistar@$\Pi^{*}$ construction}%  
\index{Main Theorem!exposition|{}}%
```

after the line

```
\section{The $\Pi^{*}$ construction}\label{S:P*}
```

Command 4

Retype the line

See also Ernest~T. Moynahan~\cite{eM57a}.

as follows:

See also Ernest~T.

\index{Moynahan, Ernest~T.}%

Moynahan~\cite{eM57a}.

Commands 5 to 7

Type the three index items

\index{lattice}%

\index{lattice!distributive}%

\index{lattice!distributive!complete}%

before the line

\begin{theorem}\label{T:P*}

Command 8

Type

\index{Main Theorem!exposition|)}

after the line

hence $\Theta = \iota$.

Command 9

Retype the line

\bibitem{sF90}

as follows:

\bibitem{sF90}\index{Foo, Soo-Key}%

Command 10

Retype the line

\bibitem{gM68}

as follows:

\bibitem{gM68}\index{Menuhin, George~A.}%

Command 11

Retype the line

```
\bibitem{eM57}
```

as follows:

```
\bibitem{eM57}\index{Moynahan, Ernest~T.}%
```

Command 12

Retype the line

```
\bibitem{eM57a}
```

as follows:

```
\bibitem{eM57a}\index{Moynahan, Ernest~T.}%
```

These `\index` commands produce the index for the `art1ind.tex` article shown in Figure 18.1. Notice that although you typed 12 index commands, only 11 entries appear in the index. The last two entries for Moynahan (commands 11 and 12) occur on the same typeset page, so only one page number shows up in the index.

INDEX

Foo, Soo-Key, 2

lattice, 1

- distributive, 1
- complete, 1

Main Theorem, 1

- exposition, 1–2

Menuhin, George A., 2

Moynahan, Ernest T., 1, 2

Π^* construction, 1

Figure 18.1: A simple index.

The `showidx` package lists all the index items of a page in a top corner on the margin. The top of the first page of the typeset `art1ind.tex` is shown in Figure 18.2.

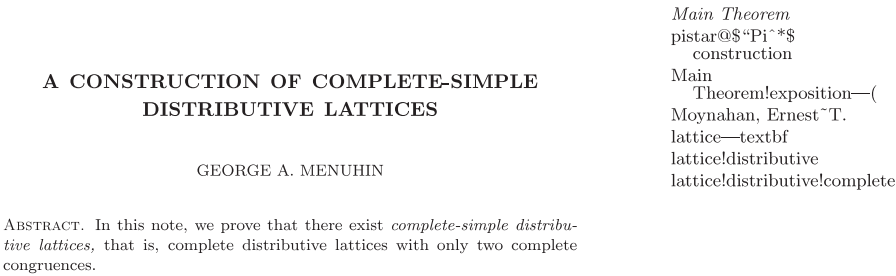


Figure 18.2: Using showidx.

18.2 Index commands

There are a few major forms of `\index` commands. They are discussed in this section, illustrated by the commands shown in Section 18.1.

Simple `\index` commands

The index entry

┌
└ Foo, Soo-Key, 2

was created by command 9,

`\index{Foo, Soo-Key}`

This entry is an example of the simplest form of an index command:

`\index{entry}`

The entry

┌
└ lattice, 2

was created as command 5,

`\index{lattice}`

Ignore, for the time being, the `\textbf` part. This entry has a subentry,

┌
└ lattice, 2
└ distributive, 2

which was created by command 6,

`\index{lattice!distributive}`

There is also a subsubentry,

```
┌
lattice, 2
    distributive, 2
└    complete, 2
```

which was created by command 7,

```
\index{lattice!distributive!complete}
```

The form of the `\index` command for subentries is

```
\index{entry!subentry}
```

and for subsubentries it is

```
\index{entry!subentry!subsubentry}
```

Modifiers

Command 5

```
\index{lattice}
```

produces a bold page number in the entry lattice.

The command whose name follows the symbol `|` (in this case, the command name is `textbf`) is applied to the page number. For instance, if you want a large bold page number, then define the command `\LargeBold` as

```
\newcommand{\LargeBoldB}[1]{\textbf{\Large #1}}
```

and type the `\index` command as

```
\index{entry|LargeBold}
```

You can also modify `\index` commands to indicate *page ranges*:

```
┌
Main Theorem, 1
└    exposition, 1–2
```

The latter index entry has a page range. It was created with commands 3 and 8:

```
\index{Main Theorem!exposition|()}
\index{Main Theorem!exposition|)}
```

Separate an entry from its modifier with `|`, open the page range with `(`, and close it with `)`.

Modifiers can also be combined. The index commands

```
\index{Main Theorem!exposition|()}
\index{Main Theorem!exposition|)}
```

produce a bold page range.

Sorting control

Observe the `\index` command

```
\index{pistar@$\Pi^{\ast}$ construction}
```

This produces the entry

```
┌
└ Π* construction, 1
```

To place this entry in the correct place in the index, use a *sort key*. The general form of an `\index` command with a sort key is

```
\index{sortkey@entry}
```

In this example, the sortkey is `pistar`. When the entries are sorted, the *sortkey* is used to sort the entry. A few typical examples follow:

Example 1 An `\index` command for G.I. Žitomirskiĭ,

```
\index{Zitomirskii@v{Z}itomirski\u{\i}, G.I.}
```

sorts Žitomirskiĭ with the Z entries.

If you used the command

```
\index{v{Z}itomirski\u{\i}, G.I.}
```

Žitomirskiĭ would be sorted with the v's.

Example 2 An `\index` command for the Örmester lemma,

```
\index{Ormester@H{O}rmester lemma}
```

would sort Örmester lemma with the O entries.

If you used the command

```
\index{H{O}rmester lemma}
```

Örmester lemma would be sorted with the H's.

Example 3 An `\index` command for *truncated* lattice,

```
\index{truncated lattice@\emph{truncated} lattice}
```

sorts *truncated* lattice with the *t* entries.

If you use the command

```
\index{\emph{truncated} lattice}
```

this would sort *truncated* lattice with the *e*'s.

Example 4 We want to place the symbol Truncat *f*, typed as `\Trunc f` (see Section 15.1.6) in the index, sorted as Trunc.

```
\index{$\Trunc f$}
```

would place Truncat *f* near the beginning of the index, sorted with the *\$* symbol. If you use the command

```
\index{Trunc@$\Trunc f$}
```

this would sort Truncat *f* with the *T*'s.

Sorting control and subentries

If you want to place a subentry under an entry with a sort key, you must include the sort key part of the entry as well:

```
\index{sortkey@entry!subentry}
```

For instance,

```
\index{Zitomirskii@\v{Z}itomirski\u{i}, G.I.!education}
```

You can also use a sort key for subentries (and subsubentries), such as

```
\index{lattice!weakly distributive@
      \emph{weakly} distributive}
```

or, a more complicated example,

```
\index{Zitomirskii@\v{Z}itomirski\u{i}, G.I.!elementary
education@\textbf{elementary} education}
```

Special characters

Since the `!`, `@`, and `|` characters have special meanings within an `\index` command, you need to *quote* those characters if you want them to appear as themselves. *MakeIndex* uses the double quote character (`"`) for this purpose: `"!`, `"@`, and `"|`.

Because this usage makes the double quote a special character itself, it also has to be quoted if you need to use it in an `\index` command: `"`.

Example 1 To produce the entry `Start here!`, type the `\index` command as

```
\index{Start here"!}
```

Example 2 To produce the entry `@ symbol`, type the `\index` command as

```
\index{"@ symbol}
```

Example 3 To produce the entry `|A|`, type the `\index` command as

```
\index{"|A"|@$"|A"|$}
```

Cross-references

It is easy to make a cross-reference to another index entry. For instance, to list distributive lattice by cross-referencing it to `lattice`, `distributive`, the command is

```
\index{distributive lattice|seeonly{lattice,
                                distributive}}
```

which produces the entry

```
┌
└ distributive lattice, see lattice, distributive
```

For non-AMS document classes, `seeonly` should be `see`.

A command of this form can be placed anywhere in the document.



Tip Put all cross-referencing `\index` commands in one place in your document, so they are easy to keep track of.

Placement of \index commands

The principle is simple.

**Rule: Placement of \index commands**

An \index command should:

1. Reference the correct page
 2. Not introduce unwanted space into the typeset document
-

For example, you should avoid placing \index commands as shown here:

```
Let $L$ be a distributive lattice
\index{lattice}
\index{distributive lattice}
that is strongly complete.
```

This placement may result in unwanted extra space following the word lattice:

```
┌
└ Let  $L$  be a distributive lattice that is strongly complete.
```

Note the placement of the \index commands in Section 18.1. In each case I have placed them as close to the referenced item as I could. If you place an index entry on a separate line, use % to comment out unwanted spaces including the end-of-line character (see Section 3.5.1), as in

```
Let $L$ be a distributive lattice
\index{lattice}%
\index{distributive lattice}%
that is strongly complete.
```

Read also Section 19.4 on page breaks and index entries.

Listing the forms of the \index command

We have discussed the following forms:

```
\index{entry}
\index{entry!subentry}
\index{entry!subentry!subsubentry}
\index{entry|modifier}
\index{entry|open/close modifier}
```

```
\index{sortkey@entry}
\index{sortkey@entry!subentry}
\index{sortkey@entry!subsortkey@subentry}
```

Of course, more combinations are possible; the following may be the longest form:

```
\index{sortkey@entry!subsortkey@subentry
!subsubsortkey@subsubentry|open/close modifier}
```

18.3 Processing the index entries

Once you are satisfied with the `\index` commands, the index is ready to be created:

Step 1 Typeset `art1ind.tex` (see Figure 18.3).

Step 2 Run the *MakeIndex* application on `art1ind.idx` (see Figure 18.4).

Step 3 Typeset `art1ind.tex` again.

You find the index on page 3 of the typeset document.

Let us look at this process in detail. In step 1 (see Figure 18.3), \LaTeX creates the `art1ind.idx` file:

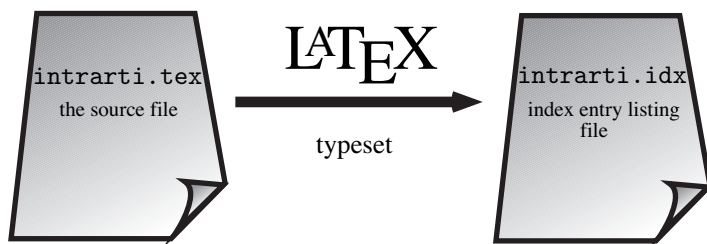
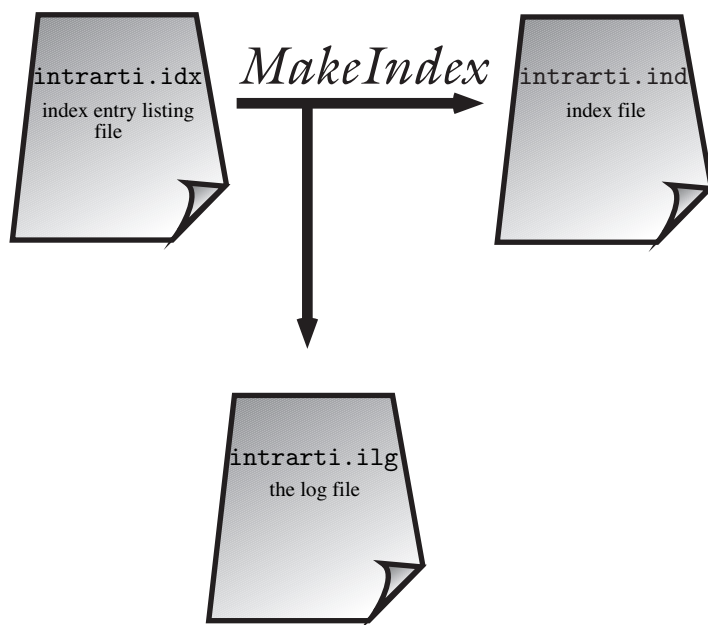
```
\indexentry{Main Theorem}{1}
\indexentry{pistar@$\Pi^{*}$ construction}{1}
\indexentry{Main Theorem!exposition|}{1}
\indexentry{Moynahan, Ernest~T.}{1}
\indexentry{lattice}{1}
\indexentry{lattice!distributive}{1}
\indexentry{lattice!distributive!complete}{1}
\indexentry{Main Theorem!exposition|){2}
\indexentry{Foo, Soo-Key}{2}
\indexentry{Menuhin, George~A.}{2}
\indexentry{Moynahan, Ernest~T.}{2}
\indexentry{Moynahan, Ernest~T.}{2}
```

In step 2 (see Figure 18.4), *MakeIndex* processes `art1ind.idx` and creates the index file `art1ind.ind`, which contains a `theindex` environment with all the index entries:

```
\begin{theindex}

\item Foo, Soo-Key, 2

\indexspace
```

Figure 18.3: Using *MakeIndex*, step 1.Figure 18.4: Using *MakeIndex*, step 2.

```

\item lattice, \textbf{1}
  \subitem distributive, 1
    \subsubitem complete, 1

\indexspace

\item Main Theorem, 1
  \subitem exposition, 1--2

```

```

\item Menuhin, George~A., 2
\item Moynahan, Ernest~T., 1, 2

\indexspace

\item $\Pi^{*}$ construction, 1

\end{theindex}

```

The `\printindex` command reads `art1ind.ind` during the next typesetting cycle.

MakeIndex also produces the index log file `art1ind.ilg`:

```

This is makeindex, version 2.14 [02-Oct-2002]
(kpathsea + Thai support).
Scanning input file art1ind.idx....done
(12 entries accepted, 0 rejected).
Sorting entries....done (43 comparisons).
Generating output file art1ind.ind....done
(22 lines written, 0 warnings).
Output written in art1ind.ind.
Transcript written in art1ind.ilg.

```

It is important to understand that in step 1, \LaTeX does not process the index entries, it simply writes the arguments of the `\index` commands in the source file to the `idx` file as arguments of `\indexentry` commands verbatim (that is, with no change). *MakeIndex* then processes the `idx` file by removing the double quote marks for the special characters, sorting the entries, and collating the page numbers. The resulting `ind` file is a normal \LaTeX source file (you can edit it, if necessary) that is included in the original document by the `\printindex` command the next time you run \LaTeX .

In Step 2, we run the *MakeIndex* application on `art1ind.idx`. How we do this depends on the \LaTeX installation you have. In UNIX installations, you type

```
MakeIndex art1i
```

at the command line. In newer user interfaces, *MakeIndex* is represented by an icon, and you drop `art1ind.idx` into it. In modern installations, your editor also runs *MakeIndex*.

18.4 Rules

There are some simple rules to keep in mind when entering index items.



Rule: Spaces in `\index`

Do not leave unnecessary spaces in the argument of an `\index` command.

`\index{item}`, `\index{_item}`, and `\index{item_}`

produces three different entries.

There are options that instruct *MakeIndex* to ignore such spaces, but you are better off typing the `\index` commands correctly in the first place.



Rule: Spacing rules for *MakeIndex*

L^AT_EX's text spacing rules (Section 3.2.1) do not apply. *MakeIndex* does not follow these rules when it sorts the index items. While L^AT_EX ignores spaces, *MakeIndex* does not.



Rule: Sort keys

In `\index{sortkey@item}`, the *sortkey* is both space and case sensitive.

For instance,

`\index{alpha@α}`

`\index{Alpha@α}`

`\index{ALPHA@α}`

represent three different items.



Rule: Braces

In every entry, the braces must be balanced.

Normally, balancing braces is not a problem. The braces within a math formula or a T_EX expression should always be balanced. However, the `\index` command that creates the entry for `{` with the sort key `leftbrace` cannot be typed as

`\index{@\{}`

because L^AT_EX would give the error message

Runaway argument?

`{leftbrace@\{}`

! Paragraph ended before \@wrindex was complete.

There are many ways to correct this `\index` command. Perhaps the simplest is to define

```
\newcommand{\printleftbrace}{\{}
```

and rewrite the `\index` command

```
\index{@\printleftbrace}
```

This produces the entry

```
┌
└ {, 1
```

There is, of course, a lot more to *MakeIndex* than what we have discussed in this short introduction, but what we have covered here should do for most documents. See Pehong Chen and Michael A. Harrison [5] for more detail. Section 14.2 of LC3 covers *MakeIndex* in great detail, including the customization of indexes.

18.5 Multiple indexes

We do multiple indexes with Enrico Gregorio's `imakeidx` package.

As an example, we split the index file into two parts, Subject Index and Author Index. We include the following three lines in the preamble:

```
\usepackage{imakeidx}
\makeindex[title=Subject Index]
\makeindex[name=author, title=Author Index]
```

Now enter each index command for authors in the following form:

```
\index{author}{Moynahan, Ernest~T.}
```

The other index commands are entered as usual.

Where you want the two indexes to appear in the typeset document, enter the commands

```
\printindex[author]
\printindex
```

These two commands are usually, but not necessarily, adjacent.

If you want to typeset the indexes in the style provided by the `theindex` environment in your document class, invoke `imakeidx` with

```
\usepackage[original]{imakeidx}
```

Also note the very useful `\indexprologue` command. It provides text (the argument of the command) to be typeset between the index header and the entries.

The `imakeidx` package has lots of nice features; see the excellent documentation.

18.6 *Glossary*

Using the glossary commands is very similar to using the corresponding index commands.

Instead of the

`index` `makeindex`

commands, use the

`glossary` `makeglossary`

commands, respectively. Glossary entries are written in the `glo` file, which corresponds to the `idx` file. \LaTeX gives you no further assistance in making a glossary file. There is no `\printglossary` command, the `glossary` environment, or *MakeGlossary* application. Thomas Henlich's `makeglos` and Nicole Talbot `glossaries` package fill the gap.

18.7 *Index styles and packages*

Style files are available in a very limited form also for *MakeIndex*. Google

`makeindex style`

for an up-to-date listing.

Indexing is a very complex process, so it is not surprising that there are many index packages available. CTAN lists a number of them in the `index` directory. The best known is `xindy`, described in detail in Section 14.4 of LC3.



Books in L^AT_EX

Since the introduction of L^AT_EX, the visual quality of articles published in mathematical journals has improved dramatically. Unfortunately, the same cannot be said of books published using L^AT_EX. A record number of very ugly books have appeared.

It is easy to understand why. While `amsart` has been designed to produce high-quality printed output, the standard book document classes do not produce attractive books without additional work.

L^AT_EX provides the `book` and the `amsbook` document classes to serve as foundations for well-designed books. Better quality books have to use document classes designed by professionals. We briefly discuss logical and visual design in Section 2.3.

So this chapter is not about how to produce a finished book using L^AT_EX. Our goal is much more modest: how to prepare a book manuscript for your publisher. In Section 19.1, we describe the book document classes `book` and `amsbook`. The table of contents and lists of figures and tables are discussed in Section 19.2.

Section 19.3 covers logical design. Section 19.4 deals with the final preparation of your edited manuscript for your publisher. Finally, Section 19.5 suggests a few more things to do if you typeset your book yourself.

19.1 Book document classes

In this section, we briefly discuss the way in which book and amsbook, the two standard book document classes, differ from the corresponding article document classes.

19.1.1 Sectioning

Book document classes have chapters, invoked with the `\chapter` command and *parts*, invoked with `\part`. The `\part` command is generally used to group chapters in longer documents, for instance in this book. Parts have no effect on the numbering of chapters, sections, and so on, so Chapter 1 of Part I is not numbered as I.1 but as 1.

Both `\chapter` and `\part` take a title as an argument, but the `\chapter` command also has an optional argument:

```
\chapter[short_title]{title}
```

The optional *short_title* argument is used in the running head. You may need to protect any fragile commands in *title* and *short_title* with the `\protect` command (see Section 3.3.3).

Here is the whole hierarchy:

```
\part
\chapter
  \section
    \subsection
      \subsubsection
        \paragraph
          \subparagraph
```

Book document classes, as a rule, do not number subsubsections or any of the sectioning divisions below that level.

Equations in chapters

By default, equations are numbered from 1 within chapters. So in Chapter 1 as well as in Chapter 3, the equations are numbered (1), (2), and so forth. If you have the

```
\numberwithin{equation}{chapter}
```

command in the preamble, then equations in Chapter 2 are numbered as (2.1), (2.2), and so on.

19.1.2 Division of the body

The book document classes formalize the division of the body into three parts:

Front matter The material that appears in the front of the document, including the title pages (normally four), table of contents, preface, introduction, and so on. \LaTeX numbers these pages using roman numerals. The front matter is introduced with the `\frontmatter` command.

Main matter The main part of the book, including the appendices if any. Page numbering starts from 1 using arabic numerals. The main matter is introduced with the `\mainmatter` command.

Back matter Material that appears in the back of the book, including the bibliography, index, and various other sections, such as the colophon,¹ afterword, and so on. The back matter is introduced with the `\backmatter` command.

For the book document class—and the document classes built on it—in the front and back matter, the `\chapter` command does not produce a chapter number but the title is listed in the table of contents. So you can start your introduction with

```
\chapter{Introduction}
```

Within such a chapter, you should use the *-ed forms of the sectioning commands `\section`, `\subsection`, and so on, otherwise you have sections with numbers such as 0.1.

In the main matter, the `\appendix` command marks the beginning of the appendices. Each subsequent chapter becomes a new appendix. For example,

```
\appendix
\chapter{A proof of the Main Theorem}\label{A:Mainproof}
```

produces an appendix with the given title.

Note that appendices may be labeled and cross-referenced. In Appendix A, sections are numbered A.1, A.2, and so on, subsections in A.1 are numbered A.1.1, A.1.2, and so on. The precise form these numbers take depends, of course, on the document class, packages, and user-specific changes (see Section 15.5.1).

The `\chapter` and `\chapter*` commands always produce a title listed in the table of contents for the `amsbook` document class, and the document classes built on it.

The following two questions are frequently asked:

My book has only one appendix. How can I get it to be called just “Appendix”, not “Appendix A”?

The single appendix in my book is being labeled “Appendix A”. How can I change this to just “Appendix”?

These questions are answered in the author FAQ of the AMS, go to

<http://www.ams.org/authors/author-faq.html>

¹The ‘finishing touch’, is a brief statement containing information about the publication.

19.1.3 Document class options

The options and defaults for the book document classes are the same as those of other document classes (see Section 11.5) with a few exceptions.

Two-sided printing

Options: `twoside` *default*
`oneside`

The `twoside` option formats the output for printing on both sides of a page.

Titlepage

Options: `titlepage` *default*
`notitlepage`

The `titlepage` option creates a separate title page. The `notitlepage` option creates no separate pages.

Chapter start

Options: `openright` *default*
`openany`

A chapter always starts on a new page. The book document class—and the document classes built on it—uses the option `openright` to start each chapter on an odd page, while the option `openany` starts each chapter on the first available new page. If you use the default option, end each chapter with the command

```
\cleardoublepage
```

Then if a chapter ends on an odd page, a blank page is added with no header or page number. The `\cleardoublepage` command is correctly coded if you use `amsbook`. Otherwise, use the package `cleardoublepage.sty` (in the `samples` folder).

The `amsbook` document class—and the document classes built on it—automatically clears to a right-hand page and leaves a totally blank page if needed.

19.1.4 Title pages

The book document class supports the commands: `\title`, `\author`, `\date`, and `\maketitle`. The `amsbook` document class supports the same commands as `amsart` (see Section 11.2), except for `\date`.

Title pages for published books should be created by a book designer for the publisher, who will also add a customized copyright page that will not be part of your submitted source file. In other contexts, you can design your own title page within the `titlepage` environment, which does not require the use of the `\maketitle` command.

19.2 Tables of contents, lists of tables and figures

A long document, as a rule, has a table of contents. It may also include a list of figures and a list of tables.

19.2.1 Tables of contents

What goes into the table of contents?

For the `amsbook` document class, and related document classes, the table of contents contains all of the titles of the various divisions associated with sectioning commands, down to the level specified by the `tocdepth` counter, as described in the last subsection of Section 15.5.1. This includes the titles of `*`-ed sections (though some document classes will omit those titles). For instance, if `tocdepth` is set to 2, then subsection titles will be included but subsubsection titles will not—this is the default setting.

The titles used in the table of contents are the full titles, not the short titles. This leaves us with the problem, . . .

This leaves us with the problem, what do we do if the title is too long? You cannot break the line with `\\`, because this would appear in table of contents. The AMS coded the following solution: enter the line break in the form

```
\except{toc}{\linebreak}
```

For the `book` document class—and the document classes built on it—the title or optional argument of the sectioning commands, subject to the value of the `tocdepth` counter, with the following exceptions:

- In Section 10.4.1, we discuss the `*`-ed versions of sectioning commands. They are excluded from the table of contents.
- If the sectioning command has a short title, then it is the short title that is utilized. The example in Section 11.2 shows why this is important. If you have `\\` in the title, you must have a short title without it, otherwise the `linebreak` would show up in the running head and the table of contents.

When you typeset your document with a table of contents, \LaTeX creates a file with the `toc` extension. The next time the document is typeset, the `toc` file is typeset too and included in your typeset document at the point where the command

```
\tableofcontents
```

appears in the source file, normally in the front matter. If your source file is named `myart.tex`, the `toc` file is named `myart.toc`. This file lists all the sectioning units as well as their titles and page numbers.

If you already have a `toc` file, the `\tableofcontents` command typesets a table of contents using the previously created `toc` file and creates a new `toc` file.

L^AT_EX adds a line to the table of contents, formatted like a section title, if you include the command

```
\addcontentsline{toc}{section}{text_to_be_added}
```

in your source file. There are three arguments:

1. The first argument informs L^AT_EX that a line, the third argument, should be added to the toc file.
2. The second argument specifies how the line should be formatted in the table of contents. In our example, the second argument is `section`, so the line is formatted as a section title in the table of contents. The second argument must be the name of a sectioning command.
3. The third argument is the text to be added.

You can add an unformatted line to the table of contents with the command

```
\addtocontents{toc}{text_to_be_added}
```

Such a command can also be used to add vertical spaces into the table of contents. For instance, if you want to add some vertical space before a part, you should insert the following line before the sectioning command for the part:

```
\addtocontents{toc}{\protect\vspace{10pt}}
```



Tip If you have a `\addcontentsline` or `\addtocontents` command in a file that is `\include-ed`, then place it as a first line of this file.

The toc file is easy to read. The following are typical lines from the table of contents file for a document using the book document class:

```
\contentsline{section}{\numberline {5-4.}Top matter}{119}
\contentsline{subsection}{\numberline {5-4.1.}
Article info}{119}
\contentsline {subsection}{\numberline {5-4.2.}
Author info}{121}
```

Section 15.5.1 explains how you can specify which levels of sectioning appear in the table of contents. Section 2.3 of LC3 lists the style parameters for the table of contents. It also shows you how to define new toc-like files and use multiple tables of contents in a single document, for instance, adding a mini table of contents for each chapter.



Tip You may have to typeset the document three times to create the table of contents and set the numbering of the rest of the document right.

1. The first typesetting creates the toc file.
2. The second inserts the table of contents with the old page numbers into the typeset document, re-records in the aux file the page numbers, which may have changed as a result of the insertion, and cross-references in the aux file, and generates a new toc file with the correct page numbers.
3. The third typesetting uses these new aux and toc files to typeset the document correctly and creates a new toc file.

Fragile commands in a movable argument, such as a section (short) title, must be `\protect`-ed (see Section 3.3.3). Here is a simple example using the table of contents. If the document contains the `\section` command

```
\section{The function \(\ f(x^{2}) \)}
```

the section title is stored in the toc file as

```
\contentsline {section}{\numberline
{1}The function\relax $ f(x^{2}) \relax \GenericError { }
{LaTeX Error: Bad math environment delimiter}{Your
command was ignored.\MessageBreak Type I <command>
<return> to replace it with another command,\MessageBreak
or <return> to continue without it.}}{1}
```

and the log file contains the message Error messages usually refer to a line in the source file, but in this case the error message refers to a line in the toc file.

The correct form for this section title is

```
\section{The function \protect\(\ f(x^{2}) \protect\)}
```

or, even simpler,

```
\verb+\section{The function $f(x^{2})$}
```

Note that this example is merely an illustration of unprotected fragile commands in movable arguments. As a rule, avoid using formulas in (sectioning) titles.

19.2.2 Lists of tables and figures

If you place a `\listoftables` command in the document, \LaTeX stores information for the list of tables in a `lot` file. The list of tables is inserted into the body of your document at the point where the command appears, normally in the front matter, following the table of contents.

A list of figures, similar to a list of tables, can be compiled with the command `\listoffigures`. This command creates an auxiliary file with the extension `lof`.

An optional argument of the `\caption` commands in your tables and figures can replace the argument in the list of tables and figures. Typically, the optional argument is used to specify a shorter caption for the list of tables or list of figures. There are other uses. For instance, you may notice that, as a rule, captions should be terminated by periods. If in the list of tables or list of figures, your book style fills the space between the text and the page number with dots, the extra period looks bad. This problem goes away if you use the following form of the `\caption` command:

```
\caption[title]{title.}
```

There are analogs of the table of contents commands for use with tables and figures. The command

```
\addtocontents{lot}{line_to_add}
```

adds a line to the list of tables or to the list of figures with the first argument `lof`.

19.2.3 Exercises

For exercises, `amsbook` provides the `xcb` environment. It is used for a series of exercises at the end of a section or chapter. The argument of the environment specifies the phrase (such as `Exercises`) to begin the list:

```
\begin{xcb}{Exercises}
\begin{enumerate}
\item A finite lattice  $L$  is modular iff it does not
contain a pentagon.\label{E:pent}
\item Can the numbers of covering pairs in\label{E:incr}
Exercise~\ref{E:pent} be increased?
\end{enumerate}
\end{xcb}
```

which typesets as

┌

Exercises

- (1) A finite lattice L is modular iff it does not contain a pentagon.
- (2) Can the numbers of covering pairs in Exercise 1 be increased?

└

19.3 Logical design

The discussion of logical and visual design in Section 2.3 applies to books even more than to articles. Since books are long and complex documents, errors in the logical design are much harder to correct.

Let us review some common sense rules.



Rule: Stick with the sectioning commands provided by the document class. Define the non-standard structures you wish to use as environments.

Here is an example which is obviously bad:

```
\vspace{18pt}

\noindent \textbf{Theorem 1.1.}
\textit{This is bad.}

\vspace{18pt}
```

And a good way to achieve the same result:

```
\begin{theorem}\label{T:Goodtheorem}
This is a good theorem.
\end{theorem}
```

The bad example leads to multiple difficulties:

- You have to number the theorems yourself. Adding, deleting, and rearranging theorems becomes difficult and updating cross-references is even harder.
- It is difficult to keep such constructs consistent.
- If the publisher decides to increase the white space before and after the theorems to 20 points, finding and changing all the appropriate commands becomes a tedious and error prone task.



Tip Define frequently used constructs as commands.

Rather than

```
\textbf{Warning! Do not exceed this amount!}
```

define

```
\newcommand{\important}[1]{\textbf{#1}}
```

and type your warnings as

```
\important{Warning! Do not exceed this amount!}
```

You or your editor can then change all the warnings to a different style with ease.



Tip Avoid text style commands.

If you use small caps for acronyms, do not type

```
\textsc{ibm}
```

but rather define

```
\newcommand{\ibm}{\textsc{ibm}}
```

and then

```
\ibm
```

or more generally

```
\newcommand{\acronym}[1]{\textsc{#1}}
```

and then

```
\acronym{ibm}
```



Tip Avoid white space commands.

Occasionally, you may feel that there should be some white space separating two paragraphs, so you do the following:

```
paragraph 1
```

```
\medskip
```

```
paragraph 2
```

It would be better to define a new command, say `\separate`, as

```
\newcommand{\separate}{\medskip}
```

and type the previous example as

```
paragraph 1
```

```
\separate
```

```
paragraph 2
```

Now such white space can be adjusted throughout the entire document by simply redefining one command. Note that redefining `\medskip` itself may have unintended side effects:

- Many environments depend on L^AT_EX’s definition of `\medskip`.
- You may have used `\medskip` in other situations as well.

Here is a short list of commands should not redefine:

<code>\bigskip</code>	<code>\hfil</code>	<code>\hspace</code>	<code>\parskip</code>	<code>\vfill</code>	<code>\vspace</code>
<code>\break</code>	<code>\hfill</code>	<code>\kern</code>	<code>\smallskip</code>	<code>\vglue</code>	
<code>\eject</code>	<code>\hglue</code>	<code>\medskip</code>	<code>\vfil</code>	<code>\vskip</code>	

19.4 *Final preparations for the publisher*

Throughout this book, there are a number of “don’ts”. Most are practices you should avoid while writing articles. When writing a book, it is even more important not to violate these rules.

When the editors, including the copy editor, are finished with your manuscript and you have the document class designed for the book, then you can start on the final preparations.

Step 1 ■ Eliminate all T_EX commands.

T_EX commands, that is, Plain T_EX commands that are not part of L^AT_EX (not listed as L^AT_EX commands in the index of this book) may interfere with L^AT_EX in unexpected ways. They may also cause problems with the style file that is created for your book.

Step 2 ■ Collect all your custom commands and environments together in one place, preferably in a separate command file (see Section 15.3).

Step 3 ■ Make sure that custom commands for notations and custom environments for structures are used consistently throughout your document.

This book uses the command `\doc` for document names, so `art1` is typed as `\doc{art1}`. Of course, `\texttt{art1}` gives the same result, but if you intermix `\doc{art1}` and `\texttt{art1}` commands, you lose the ability to easily change the way document names are displayed.

Step 4 ■ Watch out for vertical white space adding up.

- Do not directly follow one displayed math environment with another. Multiple adjacent lines of displayed mathematics should all be in the same environment.
 - If your style file uses interparagraph spacing, avoid beginning paragraphs with displayed math.
-

For instance,

```
\[
    x=y
\]
\[
    x=z
\]
```

is wrong. Use, instead, an `align` or `gather` environment.

Step 5 ■ If possible, do not place “tall” mathematical formulas inline. All formulas that might change the interline spacing, as a rule, should be displayed.

You can find examples on pages [155](#) and [410](#). Here is one more example, double hat accents used inline: $\hat{\hat{A}}$.

Step 6 ■ Read the `log` file.

- Watch for line-too-wide warnings (see Section [1.4](#)).
 - Check for font substitutions (see Section [5.5](#)).
-

Adobe Acrobat Professional has a preflight utility. It will check whether the PDF version of your typeset document has all the fonts it requires.

Step 7 ■ Do not assume that gray boxes or color illustrations appear when published exactly the way that they look on your monitor or printer.

Color work requires calibration of monitors and printers. It is often best left to the experts at the publisher.

Step 8 ■ Do not assume that the application that created your PDF files (see Section 10.4.3) can create high-quality PDF files.

Many applications can create PDF files or convert files to PDF format. Very few do it right. Ask your publisher what applications they recommend.

Font substitutions can also cause problems:

- A font that was used in typesetting the document may not be the font you intended. Missing fonts are substituted and the substitute fonts are rarely satisfactory.
- A special trap: Your publisher may have more, or maybe fewer, fonts than you do! As a result, the font substitutions on your publisher's system may be different from those on yours. Make sure that the fonts you use are not substituted.

19.5 *If you create the PDF file for your book*

Many publishers take your manuscript, prepared as described in Section 19.4, and guide it through the final steps for printing. Some books, however, are prepared by the authors for printing using a custom document class for books and submitted to the publisher as PDF files. If your book falls into this category, there are a few more things you should do before you create the final PDF file for your book.

Adjust the pages

Make sure that you are satisfied with the way the document is broken into pages by L^AT_EX and with the placement of the `figure` and `table` environments (see Section 10.4.3). If necessary, you should make last-minute changes to adjust page breaks. You may find the `\enlargethispage` command (see Section 3.6.3) very helpful at this stage. Just be sure to apply it on both facing pages.

To ensure that:

- page numbers in the index are correct,
- `\pageref` references (see Section 10.4.2) are correct,
- marginal comments (see Section 3.8.4) are properly placed,
- tables and figures are properly placed,

insert page breaks where necessary. Where pages break, add the three commands. `\linebreak`, `\pagebreak`, and `\noindent`. Here is an example. The bottom of page 3 and the top of page 4 of my book *General Lattice Theory* [17] are shown in Figure 19.1.

Now let us assume that we have to manually do this page break because some index items attached to this paragraph generate incorrect page numbers. The paragraph split by the page break is

In other words, lattice theory singles out a special type of poset for detailed investigation. To make such a definition worthwhile, it must be shown that this class of posets is a very useful class, that there are many such posets in various branches of mathematics (analysis, topology, logic, algebra, geometry, and so on), and that a general study of these posets will lead to a better understanding of the behavior of the examples. This was done in the first edition of G.~Birkhoff's `\emph{Lattice Theory} \cite{gB40}`. As we go along, we shall see many examples, most of them in the exercises. For a general survey of lattices in mathematics; see G.~Birkhoff `\cite{gB67}` and H.~H.~Crapo and G.-C.~Rota `\cite{CR70}`.

When typesetting this paragraph, L^AT_EX inserts a page break following

This was done in the first edition of G.~Birkhoff's+.

So we edit four lines as follows:

```
understanding of the behavior of the examples.
This was done in the first edition of G.~Birkhoff's
\linebreak

\pagebreak

\noindent \emph{Lattice Theory} \cite{gB40}. As we go
along, we shall see many examples, most of them in the
```

This change does not affect the appearance of the typeset page, but now pages 3 and 4 are separated by a `\pagebreak`. Make sure that any `\index` or `\label` commands are moved to the appropriate half of the paragraph. Now all index commands generate the correct page numbers.

Of course, if the page break is between paragraphs, only the `\pagebreak` command is needed. If the break occurs in the middle of a word, use `\-\linebreak` to add a hyphen.

This method works about 95 percent of the time. Occasionally, you have to drop either the `\linebreak` or the `\pagebreak` command.

In other words, lattice theory singles out a special type of poset for detailed investigation. To make such a definition worthwhile, it must be shown that this class of posets is a very useful class, that there are many such posets in various branches of mathematics (analysis, topology, logic, algebra, geometry, and so on), and that a general study of these posets will lead to a better understanding of the behavior of the examples. This was done in the first edition of G. Birkhoff's

4 I. First Concepts

Lattice Theory [1940]. As we go along, we shall see many examples, most of them in the exercises. For a general survey of lattices in mathematics, see G. Birkhoff [1967] and H. H. Crapo and G.-C. Rota [1970].

Figure 19.1: A page break.

Check for missing fonts and other defects

Open the PDF file of your book in Adobe Reader (or even better, in Adobe Acrobat Pro). Under **File**, go to **Properties...** and click on the **Fonts** tab. You will find a long list of fonts. Each one should be marked **Embedded Subset**. If all your fonts are embedded, you are in good shape.

Adobe Acrobat Pro has an excellent set of utilities to check whether your PDF file is ready for printing. You find them under **Preflight**. Adobe Acrobat Pro will correct all the mistakes it finds in the file and presents a detailed report.

Other adjustments

- Move the `figure` and `table` environments (see Section 10.4.3) physically close to where they appear in the typeset version, and change the optional argument of the `figure` and `table` environments to `!h`.
- Balance the white space on each page as necessary.
- Generate the index only after the page breaks are fixed.

Polish the auxiliary files

- Typeset the document one last time and then place the `\nofiles` command in the preamble to make sure that the auxiliary files are not overwritten.
- Normally, you should not have to edit the table of contents (`toc`) file or the `lot` and `lof` files (see Section 19.2) and your style file should take care of the formatting. Sometimes, however, an unfortunate page break makes editing necessary. In an

appropriate place, you may want to add to the text the command

```
\addtocontents{toc}{\pagebreak}
```

to avoid such edits.

- Create the index (ind) file from the new aux file, as described in Section 18.3. A lot of help is available in *The Chicago Manual of Style*, 16th edition [6] and in H. W. Fowler, *A Dictionary of Modern English Usage*. [10]; they have sections on bad breaks, remedies, and *Continued* lines in the index. Break the ind file into pages. To minimize bad breaks, use the `\enlargethispage` command, where necessary (see Section 3.6.3). Add any *Continued* entries.

Many book document classes, including book, have two problems with the Index.

- (i) There is no Index entry in the table of contents.
- (ii) The first page of the Index is numbered.

These are easy to correct. Add to the beginning of the ind file the command

```
\thispagestyle{empty}
```

and precede the `\printindex` with

```
\addtocontents{toc}{Index}
```

Math symbol tables

A.1 Hebrew and Greek letters

Hebrew letters

Type	Typeset
<code>\aleph</code>	\aleph
<code>\beth</code>	\beth
<code>\daleth</code>	\daleth
<code>\gimel</code>	\gimel

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/978-3-031-55281-6>

Greek letters**Lowercase**

Type	Typeset	Type	Typeset	Type	Typeset
<code>\alpha</code>	α	<code>\iota</code>	ι	<code>\sigma</code>	σ
<code>\beta</code>	β	<code>\kappa</code>	κ	<code>\tau</code>	τ
<code>\gamma</code>	γ	<code>\lambda</code>	λ	<code>\upsilon</code>	υ
<code>\delta</code>	δ	<code>\mu</code>	μ	<code>\phi</code>	ϕ
<code>\epsilon</code>	ϵ	<code>\nu</code>	ν	<code>\chi</code>	χ
<code>\zeta</code>	ζ	<code>\xi</code>	ξ	<code>\psi</code>	ψ
<code>\eta</code>	η	<code>\pi</code>	π	<code>\omega</code>	ω
<code>\theta</code>	θ	<code>\rho</code>	ρ		
<code>\varepsilon</code>	ε	<code>\varpi</code>	ϖ	<code>\varsigma</code>	ς
<code>\vartheta</code>	ϑ	<code>\varrho</code>	ϱ	<code>\varphi</code>	φ
	<code>\digamma</code>	F	<code>\varkappa</code>	\varkappa	

Uppercase

Type	Typeset	Type	Typeset	Type	Typeset
<code>\Gamma</code>	Γ	<code>\Xi</code>	Ξ	<code>\Phi</code>	Φ
<code>\Delta</code>	Δ	<code>\Pi</code>	Π	<code>\Psi</code>	Ψ
<code>\Theta</code>	Θ	<code>\Sigma</code>	Σ	<code>\Omega</code>	Ω
<code>\Lambda</code>	Λ	<code>\Upsilon</code>	Υ		
<code>\varGamma</code>	\varGamma	<code>\varXi</code>	\varXi	<code>\varPhi</code>	\varPhi
<code>\varDelta</code>	\varDelta	<code>\varPi</code>	\varPi	<code>\varPsi</code>	\varPsi
<code>\varTheta</code>	\varTheta	<code>\varSigma</code>	\varSigma	<code>\varOmega</code>	\varOmega
<code>\varLambda</code>	\varLambda	<code>\varUpsilon</code>	\varUpsilon		

A.2 Binary relations

Type	Typeset	Type	Typeset
<code><</code>	$<$	<code>></code>	$>$
<code>=</code>	$=$	<code>:</code>	$:$
<code>\in</code>	\in	<code>\ni</code> or <code>\owns</code>	\ni
<code>\leq</code> or <code>\le</code>	\leq	<code>\geq</code> or <code>\ge</code>	\geq
<code>\ll</code>	\ll	<code>\gg</code>	\gg
<code>\prec</code>	\prec	<code>\succ</code>	\succ
<code>\preceq</code>	\preceq	<code>\succeq</code>	\succeq
<code>\sim</code>	\sim	<code>\approx</code>	\approx
<code>\simeq</code>	\simeq	<code>\cong</code>	\cong
<code>\equiv</code>	\equiv	<code>\doteq</code>	\doteq
<code>\subset</code>	\subset	<code>\supset</code>	\supset
<code>\subseteq</code>	\subseteq	<code>\supseteq</code>	\supseteq
<code>\sqsubseteq</code>	\sqsubseteq	<code>\sqsupseteq</code>	\sqsupseteq
<code>\smile</code>	\smile	<code>\frown</code>	\frown
<code>\perp</code>	\perp	<code>\models</code>	\models
<code>\mid</code>	\mid	<code>\parallel</code>	\parallel
<code>\vdash</code>	\vdash	<code>\dashv</code>	\dashv
<code>\propto</code>	\propto	<code>\asymp</code>	\asymp
<code>\bowtie</code>	\bowtie		
<code>\sqsubset</code>	\sqsubset	<code>\sqsupset</code>	\sqsupset
<code>\Join</code>	\Join		

Note the `\colon` command used in $f: x \rightarrow x^2$, typed as

```
f \colon x \to x^2
```

More binary relations

Type	Typeset	Type	Typeset
<code>\leqq</code>	\leqslant	<code>\geqq</code>	\geqslant
<code>\leqslant</code>	\leq	<code>\geqslant</code>	\geq
<code>\eqslantless</code>	\lessapprox	<code>\eqslantgtr</code>	\gtrapprox
<code>\lesssim</code>	\lesssim	<code>\gtrsim</code>	\gtrsim
<code>\lessapprox</code>	\lessapprox	<code>\gtrapprox</code>	\gtrapprox
<code>\approxeq</code>	\approx		
<code>\lessdot</code>	\lessdot	<code>\gtrdot</code>	\gtrdot
<code>\lll</code>	\lll	<code>\ggg</code>	\ggg
<code>\lessgtr</code>	\lessgtr	<code>\gtrless</code>	\gtrless
<code>\lesseqgtr</code>	\lesseqgtr	<code>\gtreqless</code>	\gtreqless
<code>\lesseqqgtr</code>	\lesseqqgtr	<code>\gtreqqless</code>	\gtreqqless
<code>\doteqdot</code>	\doteqdot	<code>\eqcirc</code>	\eqcirc
<code>\circeq</code>	\circeq	<code>\triangleq</code>	\triangleq
<code>\risingdotseq</code>	\risingdotseq	<code>\fallingdotseq</code>	\fallingdotseq
<code>\backsim</code>	\backsim	<code>\thicksim</code>	\thicksim
<code>\backsimeq</code>	\backsimeq	<code>\thickapprox</code>	\thickapprox
<code>\preccurlyeq</code>	\preccurlyeq	<code>\succcurlyeq</code>	\succcurlyeq
<code>\curlyeqprec</code>	\curlyeqprec	<code>\curlyeqsucc</code>	\curlyeqsucc
<code>\precsim</code>	\precsim	<code>\succsim</code>	\succsim
<code>\precapprox</code>	\precapprox	<code>\succapprox</code>	\succapprox
<code>\subseteq</code>	\subseteq	<code>\supseteq</code>	\supseteq
<code>\Subset</code>	\Subset	<code>\Supset</code>	\Supset
<code>\vartriangleleft</code>	\vartriangleleft	<code>\vartriangleright</code>	\vartriangleright
<code>\trianglelefteq</code>	\trianglelefteq	<code>\trianglerighteq</code>	\trianglerighteq
<code>\vDash</code>	\vDash	<code>\Vdash</code>	\Vdash
<code>\Vvdash</code>	\Vvdash		
<code>\smallsmile</code>	\smallsmile	<code>\smallfrown</code>	\smallfrown
<code>\shortmid</code>	\shortmid	<code>\shortparallel</code>	\shortparallel
<code>\bumpeq</code>	\bumpeq	<code>\Bumpeq</code>	\Bumpeq
<code>\between</code>	\between	<code>\pitchfork</code>	\pitchfork
<code>\varpropto</code>	\varpropto	<code>\backepsilon</code>	\backepsilon
<code>\blacktriangleleft</code>	\blacktriangleleft	<code>\blacktriangleright</code>	\blacktriangleright
<code>\therefore</code>	\therefore	<code>\because</code>	\because

Negated binary relations

Type	Typeset	Type	Typeset
<code>\neq</code> or <code>\ne</code>	\neq	<code>\notin</code>	\notin
<code>\nless</code>	\nless	<code>\ngtr</code>	\ngtr
<code>\nleq</code>	\nleq	<code>\ngeq</code>	\ngeq
<code>\nleqslant</code>	\nleqslant	<code>\ngeqslant</code>	\ngeqslant
<code>\nleqq</code>	\nleqq	<code>\ngeqq</code>	\ngeqq
<code>\lneq</code>	\lneq	<code>\gneq</code>	\gneq
<code>\lneqq</code>	\lneqq	<code>\gneqq</code>	\gneqq
<code>\lvertneqq</code>	\lvertneqq	<code>\gvertneqq</code>	\gvertneqq
<code>\lnsim</code>	\lnsim	<code>\gnsim</code>	\gnsim
<code>\lnapprox</code>	\lnapprox	<code>\gnapprox</code>	\gnapprox
<code>\nprec</code>	\nprec	<code>\nsucc</code>	\nsucc
<code>\npreceq</code>	\npreceq	<code>\nsucceq</code>	\nsucceq
<code>\precneqq</code>	\precneqq	<code>\succneqq</code>	\succneqq
<code>\precnsim</code>	\precnsim	<code>\succnsim</code>	\succnsim
<code>\precnapprox</code>	\precnapprox	<code>\succnapprox</code>	\succnapprox
<code>\nsim</code>	\nsim	<code>\ncong</code>	\ncong
<code>\nshortmid</code>	\nshortmid	<code>\nshortparallel</code>	\nshortparallel
<code>\nmid</code>	\nmid	<code>\nparallel</code>	\nparallel
<code>\nvDash</code>	\nvDash	<code>\nvDash</code>	\nvDash
<code>\nVDash</code>	\nVDash	<code>\nVDash</code>	\nVDash
<code>\ntriangleleft</code>	\ntriangleleft	<code>\ntriangleright</code>	\ntriangleright
<code>\ntrianglelefteq</code>	\ntrianglelefteq	<code>\ntrianglerighteq</code>	\ntrianglerighteq
<code>\nsubseteq</code>	\nsubseteq	<code>\nsupseteq</code>	\nsupseteq
<code>\nsubseteqq</code>	\nsubseteqq	<code>\nsupseteqq</code>	\nsupseteqq
<code>\subsetneq</code>	\subsetneq	<code>\supsetneq</code>	\supsetneq
<code>\varsubsetneq</code>	\varsubsetneq	<code>\varsupsetneq</code>	\varsupsetneq
<code>\subsetneqq</code>	\subsetneqq	<code>\supsetneqq</code>	\supsetneqq
<code>\varsubsetneqq</code>	\varsubsetneqq	<code>\varsupsetneqq</code>	\varsupsetneqq

A.3 Binary operations

Type	Typeset	Type	Typeset
$+$	$+$	$-$	$-$
$\backslash pm$	\pm	$\backslash mp$	\mp
$\backslash times$	\times	$\backslash cdot$	\cdot
$\backslash circ$	\circ	$\backslash bigcirc$	\bigcirc
$\backslash div$	\div	$\backslash bmod$	\bmod
$\backslash cap$	\cap	$\backslash cup$	\cup
$\backslash sqcap$	\sqcap	$\backslash sqcup$	\sqcup
$\backslash wedge$ or $\backslash land$	\wedge	$\backslash vee$ or $\backslash lor$	\vee
$\backslash triangleleft$	\triangleleft	$\backslash triangleright$	\triangleright
$\backslash bigtriangleup$	\bigtriangleup	$\backslash bigtriangledown$	\bigtriangledown
$\backslash oplus$	\oplus	$\backslash ominus$	\ominus
$\backslash otimes$	\otimes	$\backslash oslash$	\oslash
$\backslash odot$	\odot	$\backslash bullet$	\bullet
$\backslash dagger$	\dagger	$\backslash ddagger$	\ddagger
$\backslash setminus$	\backslash	$\backslash smallsetminus$	\smallsetminus
$\backslash wr$	\wr	$\backslash amalg$	\amalg
$\backslash ast$	$*$	$\backslash star$	\star
$\backslash diamond$	\diamond		
$\backslash lhd$	\triangleleft	$\backslash rhd$	\triangleright
$\backslash unlhd$	\trianglelefteq	$\backslash unrhd$	\trianglerighteq
$\backslash dotplus$	$\dot{+}$	$\backslash centerdot$	\cdot
$\backslash ltimes$	\ltimes	$\backslash rtimes$	\rtimes
$\backslash leftthreetimes$	\leftthreetimes	$\backslash rightthreetimes$	\rightthreetimes
$\backslash circleddash$	\ominus	$\backslash uplus$	\uplus
$\backslash barwedge$	$\bar{\wedge}$	$\backslash doublebarwedge$	\equiv
$\backslash curlywedge$	\curlywedge	$\backslash curlyvee$	\curlyvee
$\backslash veebar$	\veebar	$\backslash intercal$	\intercal
$\backslash doublecap$ or $\backslash Cap$	\mho	$\backslash doublecup$ or $\backslash Cup$	\mho
$\backslash circledast$	\circledast	$\backslash circledcirc$	\circledcirc
$\backslash boxminus$	\boxminus	$\backslash boxtimes$	\boxtimes
$\backslash boxdot$	\boxdot	$\backslash boxplus$	\boxplus
$\backslash divideontimes$	\div	$\backslash vartriangle$	\triangle
$\backslash And$	$\&$		

A.4 Arrows

Type	Typeset	Type	Typeset
<code>\leftarrow</code>	\leftarrow	<code>\rightarrow</code> or <code>\to</code>	\rightarrow
<code>\longleftarrow</code>	\longleftarrow	<code>\longrightarrow</code>	\longrightarrow
<code>\Leftarrow</code>	\Leftarrow	<code>\Rightarrow</code>	\Rightarrow
<code>\Longleftarrow</code>	\Longleftarrow	<code>\Longrightarrow</code>	\Longrightarrow
<code>\leftrightarrow</code>	\leftrightarrow	<code>\longleftrightarrow</code>	\longleftrightarrow
<code>\Leftrightarrow</code>	\Leftrightarrow	<code>\Longleftrightarrow</code>	\Longleftrightarrow
<code>\uparrow</code>	\uparrow	<code>\downarrow</code>	\downarrow
<code>\Uparrow</code>	\Uparrow	<code>\Downarrow</code>	\Downarrow
<code>\updownarrow</code>	\updownarrow	<code>\Updownarrow</code>	\Updownarrow
<code>\nearrow</code>	\nearrow	<code>\searrow</code>	\searrow
<code>\swarrow</code>	\swarrow	<code>\nwarrow</code>	\nwarrow
<code>\iff</code>	\iff	<code>\mapsto</code>	\mapsto
<code>\mapsto</code>	\mapsto	<code>\longmapsto</code>	\longmapsto
<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookleftarrow</code>	\hookleftarrow
<code>\leftharpoonup</code>	\leftharpoonup	<code>\rightharpoonup</code>	\rightharpoonup
<code>\leftharpoondown</code>	\leftharpoondown	<code>\rightharpoondown</code>	\rightharpoondown
<code>\leadsto</code>	\leadsto		
<code>\leftleftarrows</code>	\leftleftarrows	<code>\rightrightarrows</code>	\rightrightarrows
<code>\leftrightarrows</code>	\leftrightarrows	<code>\rightleftarrows</code>	\rightleftarrows
<code>\Lleftarrow</code>	\Lleftarrow	<code>\Rrightarrow</code>	\Rrightarrow
<code>\twoheadleftarrow</code>	\twoheadleftarrow	<code>\twoheadrightarrow</code>	\twoheadrightarrow
<code>\leftarrowtail</code>	\leftarrowtail	<code>\rightarrowtail</code>	\rightarrowtail
<code>\looparrowleft</code>	\looparrowleft	<code>\looparrowright</code>	\looparrowright
<code>\upuparrows</code>	\upuparrows	<code>\downdownarrows</code>	\downdownarrows
<code>\upharpoonleft</code>	\upharpoonleft	<code>\upharpoonright</code>	\upharpoonright
<code>\downharpoonleft</code>	\downharpoonleft	<code>\downharpoonright</code>	\downharpoonright
<code>\leftrightsquigarrow</code>	\leftrightsquigarrow	<code>\rightsquigarrow</code>	\rightsquigarrow
<code>\multimap</code>	\multimap		
<code>\nleftarrow</code>	\nleftarrow	<code>\nrightarrow</code>	\nrightarrow
<code>\nLeftarrow</code>	\nLeftarrow	<code>\nRightarrow</code>	\nRightarrow
<code>\nleftrightarrow</code>	\nleftrightarrow	<code>\nLeftrightarrow</code>	\nLeftrightarrow
<code>\dashleftarrow</code>	\dashleftarrow	<code>\dashrightarrow</code>	\dashrightarrow
<code>\curvearrowleft</code>	\curvearrowleft	<code>\curvearrowright</code>	\curvearrowright
<code>\circlearrowleft</code>	\circlearrowleft	<code>\circlearrowright</code>	\circlearrowright
<code>\leftrightharpoons</code>	\leftrightharpoons	<code>\rightleftharpoons</code>	\rightleftharpoons
<code>\Lsh</code>	\Lsh	<code>\Rsh</code>	\Rsh

Type	Typeset	Type	Typeset
<code>\hbar</code>	\hbar	<code>\ell</code>	ℓ
<code>\imath</code>	\imath	<code>\jmath</code>	\jmath
<code>\wp</code>	\wp	<code>\partial</code>	∂
<code>\Im</code>	\Im	<code>\Re</code>	\Re
<code>\infty</code>	∞	<code>\prime</code>	\prime
<code>\emptyset</code>	\emptyset	<code>\varnothing</code>	\varnothing
<code>\forall</code>	\forall	<code>\exists</code>	\exists
<code>\smallint</code>	\int	<code>\triangle</code>	\triangle
<code>\top</code>	\top	<code>\bot</code>	\bot
<code>\P</code>	\P	<code>\S</code>	\S
<code>\dag</code>	\dagger	<code>\ddag</code>	\ddagger
<code>\flat</code>	\flat	<code>\natural</code>	\natural
<code>\sharp</code>	\sharp	<code>\angle</code>	\angle
<code>\clubsuit</code>	\clubsuit	<code>\diamondsuit</code>	\diamondsuit
<code>\heartsuit</code>	\heartsuit	<code>\spadesuit</code>	\spadesuit
<code>\surd</code>	\surd	<code>\nabla</code>	∇
<code>\pounds</code>	\pounds	<code>\neg</code> or <code>\lnot</code>	\neg
<code>\Box</code>	\Box	<code>\Diamond</code>	\Diamond
<code>\mho</code>	\mho		
<code>\hslash</code>	\hslash	<code>\complement</code>	\complement
<code>\backprime</code>	\backprime	<code>\nexists</code>	\nexists
<code>\Bbbk</code>	\Bbbk		
<code>\diagup</code>	\diagup	<code>\diagdown</code>	\diagdown
<code>\blacktriangle</code>	\blacktriangle	<code>\blacktriangledown</code>	\blacktriangledown
<code>\triangledown</code>	\triangledown	<code>\eth</code>	\eth
<code>\square</code>	\square	<code>\blacksquare</code>	\blacksquare
<code>\lozenge</code>	\lozenge	<code>\blacklozenge</code>	\blacklozenge
<code>\measuredangle</code>	\measuredangle	<code>\sphericalangle</code>	\sphericalangle
<code>\circledS</code>	\circledS	<code>\bigstar</code>	\bigstar
<code>\Finv</code>	\Finv	<code>\Game</code>	\Game

A.5 Delimiters

Name	Type	Typeset
left parenthesis	((
right parenthesis))
left bracket	[or \lbrack	[
right bracket] or \rbrack]
left brace	\{ or \lbrace	{
right brace	\} or \rbrace	}
backslash	\backslash	\
forward slash	/	/
left angle bracket	\langle	<
right angle bracket	\rangle	>
vertical line	or \vert	
double vertical line	\ or \Vert	
left floor	\lfloor	⌊
right floor	\rfloor	⌋
left ceiling	\lceil	⌈
right ceiling	\rceil	⌉
upward	\uparrow	↑
double upward	\Uparrow	⇑
downward	\downarrow	↓
double downward	\Downarrow	⇓
up-and-down	\updownarrow	↕
double up-and-down	\Updownarrow	⇕
upper-left corner	\ulcorner	⏟
upper-right corner	\urcorner	⏟
lower-left corner	\llcorner	⏟
lower-right corner	\lrcorner	⏟

A.6 Operators

“Pure” operators, with no limits

Type	Typeset	Type	Typeset	Type	Typeset	Type	Typeset
<code>\arccos</code>	\arccos	<code>\cot</code>	\cot	<code>\hom</code>	\hom	<code>\sin</code>	\sin
<code>\arcsin</code>	\arcsin	<code>\coth</code>	\coth	<code>\ker</code>	\ker	<code>\sinh</code>	\sinh
<code>\arctan</code>	\arctan	<code>\csc</code>	\csc	<code>\lg</code>	\lg	<code>\tan</code>	\tan
<code>\arg</code>	\arg	<code>\deg</code>	\deg	<code>\ln</code>	\ln	<code>\tanh</code>	\tanh
<code>\cos</code>	\cos	<code>\dim</code>	\dim	<code>\log</code>	\log		
<code>\cosh</code>	\cosh	<code>\exp</code>	\exp	<code>\sec</code>	\sec		

Operators with limits

Type	Typeset	Type	Typeset
<code>\det</code>	\det	<code>\limsup</code>	\limsup
<code>\gcd</code>	\gcd	<code>\max</code>	\max
<code>\inf</code>	\inf	<code>\min</code>	\min
<code>\lim</code>	\lim	<code>\Pr</code>	\Pr
<code>\liminf</code>	\liminf	<code>\sup</code>	\sup
<code>\injlim</code>	injlim	<code>\projlim</code>	$\operatorname{projlim}$
<code>\varliminf</code>	\varliminf	<code>\varlimsup</code>	\varlimsup
<code>\varinjlim</code>	\varinjlim	<code>\varprojlim</code>	\varprojlim

A.6.1 Large operators

Type	Inline	Displayed
<code>\int_{a}^{b}</code>	\int_a^b	\int_a^b
<code>\oint_{a}^{b}</code>	\oint_a^b	\oint_a^b
<code>\iint_{a}^{b}</code>	\iint_a^b	\iint_a^b
<code>\iiint_{a}^{b}</code>	\iiint_a^b	\iiint_a^b
<code>\iiiiiint_{a}^{b}</code>	\iiiiiint_a^b	\iiiiiint_a^b
<code>\idotsint_{a}^{b}</code>	$\int \cdots \int_a^b$	$\int \cdots \int_a^b$
<code>\prod_{i=1}^n</code>	$\prod_{i=1}^n$	$\prod_{i=1}^n$
<code>\coprod_{i=1}^n</code>	$\coprod_{i=1}^n$	$\coprod_{i=1}^n$
<code>\bigcap_{i=1}^n</code>	$\bigcap_{i=1}^n$	$\bigcap_{i=1}^n$
<code>\bigcup_{i=1}^n</code>	$\bigcup_{i=1}^n$	$\bigcup_{i=1}^n$
<code>\bigwedge_{i=1}^n</code>	$\bigwedge_{i=1}^n$	$\bigwedge_{i=1}^n$
<code>\bigvee_{i=1}^n</code>	$\bigvee_{i=1}^n$	$\bigvee_{i=1}^n$
<code>\bigsqcup_{i=1}^n</code>	$\bigsqcup_{i=1}^n$	$\bigsqcup_{i=1}^n$
<code>\biguplus_{i=1}^n</code>	$\biguplus_{i=1}^n$	$\biguplus_{i=1}^n$
<code>\bigotimes_{i=1}^n</code>	$\bigotimes_{i=1}^n$	$\bigotimes_{i=1}^n$
<code>\bigoplus_{i=1}^n</code>	$\bigoplus_{i=1}^n$	$\bigoplus_{i=1}^n$
<code>\bigodot_{i=1}^n</code>	$\bigodot_{i=1}^n$	$\bigodot_{i=1}^n$
<code>\sum_{i=1}^n</code>	$\sum_{i=1}^n$	$\sum_{i=1}^n$

A.7 Math accents and fonts

Math accents

Type	Typeset	amsxtra	
		Type	Typeset
<code>\acute{a}</code>	\acute{a}		
<code>\bar{a}</code>	\bar{a}		
<code>\breve{a}</code>	\breve{a}	<code>\spbreve</code>	\breve{a}
<code>\check{a}</code>	\check{a}	<code>\spcheck</code>	\check{a}
<code>\dot{a}</code>	\dot{a}	<code>\spdot</code>	\dot{a}
<code>\ddot{a}</code>	\ddot{a}	<code>\spddot</code>	\ddot{a}
<code>\dddota</code>	\dddot{a}	<code>\spdddot</code>	\dddot{a}
<code>\grave{a}</code>	\grave{a}		
<code>\hat{a}</code>	\hat{a}		
<code>\widehat{a}</code>	\widehat{a}	<code>\sphat</code>	\widehat{a}
<code>\mathring{a}</code>	\mathring{a}		
<code>\tilde{a}</code>	\tilde{a}		
<code>\widetilde{a}</code>	\widetilde{a}	<code>\sptilde</code>	\widetilde{a}
<code>\vec{a}</code>	\vec{a}		

Math fonts

Type	Typeset
\LaTeX	
<code>\mathbf{A}</code>	\mathbf{A}
<code>\mathcal{A}</code>	\mathcal{A}
<code>\mathit{A}</code>	A
<code>\mathnormal{A}</code>	A
<code>\mathrm{A}</code>	A
<code>\mathsf{A}</code>	A
<code>\mathtt{A}</code>	A
<code>\boldsymbol{\alpha}</code>	$\boldsymbol{\alpha}$
<code>\mathbb{A}</code>	\mathbb{A}
<code>\mathfrak{A}</code>	\mathfrak{A}
<code>\mathscr{A}</code>	\mathscr{A}

And thousand more from STIX (see Section 8.2)!

`\mathscr` requires the `eucal` package with the `mathscr` option

A.8 *Math spacing commands*

Name	Width	Short	Long
1 mu (math unit)	μ	<code>\mspace{1mu}</code>	
thinspace	μ	<code>\,</code>	<code>\thinspace</code>
medspace	μ	<code>\:</code>	<code>\medspace</code>
thickspace	μ	<code>\;</code>	<code>\thickspace</code>
interword space	μ	<code>\quad</code>	
1 em	μ		<code>\quad</code>
2 em	μ		<code>\qquad</code>
Negative space			
1 mu	μ		<code>\mspace{-1mu}</code>
thinspace	μ	<code>\!</code>	<code>\negthinspace</code>
medspace	μ		<code>\negmedspace</code>
thickspace	μ		<code>\negthickspace</code>

Text symbol tables

B.1 Some European characters

Name	Type	Typeset	Type	Typeset
a-ring	\aa	å	\AA	Å
aesc	\ae	æ	\AE	Æ
ethel	\oe	œ	\OE	Œ
eszett	\ss	ß	\SS	ŠŠ
inverted question mark	?‘	¿		
inverted exclamation mark	!‘	¡		
slashed L	\l	ł	\L	Ł
slashed O	\o	ø	\O	Ø

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/978-3-031-55281-6>

B.2 Text accents

Name	Type	Typeset	Name	Type	Typeset
acute	\'{o}	ó	macron	\={o}	ō
breve	\u{o}	ö	overdot	\.{g}	ġ
caron/haček	\v{o}	ě	ring	\r{u}	ŭ
cedilla	\c{c}	ç	tie	\t{oo}	ôo
circumflex	\^{o}	ô	tilde	\~{n}	ñ
dieresis/umlaut	\" {u}	ü	underdot	\d{m}	ṁ
double acute	\H{o}	ő	underbar	\b{o}	ḡ
grave	\`{o}	ò			
dotless i	\i	ı	dotless j	\j	ȷ
	\' {\i}	í		\v{\j}	ȶ

B.3 Text font commands

B.3.1 Text font family commands

Command with Argument	Command Declaration	Switches to the font family
\textnormal{...}	{\normalfont ...}	document
\emph{...}	{\em ...}	<i>emphasis</i>
\textrm{...}	{\rmfamily ...}	roman
\textsf{...}	{\sffamily ...}	sans serif
\texttt{...}	{\ttfamily ...}	typewriter style
\textup{...}	{\upshape ...}	upright shape
\textit{...}	{\itshape ...}	<i>italic shape</i>
\textsl{...}	{\slshape ...}	<i>slanted shape</i>
\textsc{...}	{\scshape ...}	SMALL CAPITALS
\textbf{...}	{\bfseries ...}	bold
\textmd{...}	{\mdseries ...}	normal weight and width

B.3.2 Text font size changes

Command	L ^A T _E X sample text	AMS sample text
<code>\Tiny</code>	[not available]	sample text
<code>\tiny</code>	sample text	sample text
<code>\SMALL</code> or <code>\scriptsize</code>	sample text	sample text
<code>\Small</code> or <code>\footnotesize</code>	sample text	sample text
<code>\small</code>	sample text	sample text
<code>\normalsize</code>	sample text	sample text
<code>\large</code>	sample text	sample text
<code>\Large</code>	sample text	sample text
<code>\LARGE</code>	sample text	sample text
<code>\huge</code>	sample text	sample text
<code>\Huge</code>	sample text	sample text

B.4 Additional text symbols

Name	Type	Typeset
ampersand	<code>\&</code>	&
asterisk bullet	<code>\textasteriskcentered</code>	*
backslash	<code>\textbackslash</code>	\
bar (caesura)	<code>\textbar</code>	
brace left	<code>\{</code>	{
brace right	<code>\}</code>	}
bullet	<code>\textbullet</code>	•
circled a	<code>\textcircled{a}</code>	Ⓐ
circumflex	<code>\textasciicircum</code>	^
copyright	<code>\copyright</code>	©
dagger	<code>\dag</code>	†
double dagger (diesis)	<code>\ddag</code>	‡
dollar	<code>\\$</code>	\$
double quotation left	<code>\textquotedblleft</code> or ‘	“
double quotation right	<code>\textquotedblright</code> or ’	”
em dash	<code>\textemdash</code> or ---	—
en dash	<code>\textendash</code> or --	–
exclamation down	<code>\textexclamdown</code> or !’	¡
greater than	<code>\textgreater</code>	>
less than	<code>\textless</code>	<
lowline	<code>_</code>	-
midpoint	<code>\textperiodcentered</code>	·
octothorp	<code>\#</code>	#
percent	<code>\%</code>	%
pilcrow (paragraph)	<code>\P</code>	¶
question down	<code>\textquestiondown</code> or ?’	¿
registered trademark	<code>\textregistered</code>	®
section	<code>\S</code>	§

Additional text symbols, continued

Name	Type	Typeset
single quote left	<code>\textquoteleft</code> or ‘	‘
single quote right	<code>\textquoteright</code> or ’	’
sterling	<code>\pounds</code>	£
superscript	<code>a</code>	^a
tilde	<code>\textasciitilde</code>	~
trademark	<code>\texttrademark</code>	™
visible space	<code>\textvisiblespace</code>	␣

B.5 Additional text symbols with T1 encoding***An accent***

Name	Type	Typeset
Ogonek	<code>\k{e}</code>	ę

European characters

Name	Type	Typeset	Type	Typeset
Eth	<code>\dh</code>	ð	<code>\DH</code>	Ð
Dyet	<code>\dj</code>	đ	<code>\DJ</code>	Đ
Eng	<code>\ng</code>	ŋ	<code>\NG</code>	Ŋ
Thorn	<code>\th</code>	þ	<code>\TH</code>	Þ

Quotation marks

Name	Type	Typeset	Type	Typeset
Single Guillemet	<code>\guilsinglleft</code>	◀	<code>\guilsinglright</code>	▶
Double Guillemet	<code>\guillemotleft</code>	«	<code>\guillemotright</code>	»
Single Quotation	<code>\quotesinglbase</code>	,	<code>\textquoteright</code>	’
Double Quotation	<code>\quotedblbase</code>	„	<code>\textquotedbl</code>	”

B.6 Text spacing commands

Name	Width	Short command	Long command
Positive Space			
Normal	varies	□	
Intersentence	varies	\@.□	
Interword	varies	\□	
Italic Corr.	varies	\/\□	
Tie	varies	~	
Thinspace	␣	\,	\thinspace
Medspace	␣	\:	\medspace
Thickspace	␣	\;	\thickspace
1 em	□		\quad
2 em	□□		\qquad
Negative Space			
Thinspace	␣	\!	\negthinspace
Medspace	␣		\negmedspace
Thickspace	␣		\negthickspace

ChatGPT 101 for L^AT_EX users

C.1 Introduction

While working on this book, I discovered unexpected assistance in ChatGPT. Often, I turned to ChatGPT for LaTeX commands, seeking definitions and examples.

In my discussions and email exchanges with experts, skepticism was a common response: “You know my opinion of ChatGPT and other large language models; they can be misleading and under certain circumstances, produce entirely captivating yet nonsensical and fabricated content.”

This response puzzled me. Each time I consulted ChatGPT, it provided clear, unambiguous answers and practical BibL^AT_EX code. To enable you to gain the same benefits, I decided to write a ChatGPT primer for LaTeX users.

C.2 What is ChatGPT?

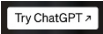
ChatGPT is an LLM (Large Language Model), an artificial intelligence model that is trained to understand and generate human language. These models are termed ‘large’ because they are trained on extensive text data—including books, articles, websites, and other digital texts—to grasp language patterns, vocabulary, and grammar.

LLMs can execute various language-related tasks, such as answering questions, completing sentences, and summarizing texts. The OpenAI GPT series (Generative Pre-trained Transformer) are notable examples of LLMs.

These models have garnered considerable attention for their ability to produce coherent and contextually pertinent text, proving invaluable in many different domains ranging from creative writing to technical support and coding.

To start using ChatGPT is remarkably easy. Go to

`openai.org`

and on the upper right corner, click on . That's it.

C.3 What it can't do

It should now be clear that we should ask ChatGPT questions that require large amount of data it acquired to answer. Otherwise you may get 'nonsensical and fabricated content'.

Donald Knuth¹ [30] illustrates it:

'Since one of today's popular recreations is to play with ChatGPT, I decided on 07 April 2023 to try my own little experiment.'

Knuth's first question:

1. Tell me what Donald Knuth says to Stephen Wolfram about ChatGPT.

Clearly, the answer you get does not make sense. There is no large relevant data set. You are asking the shepherd in New Zealand about elephants.

His tenth question is better:

10. How many parts are in The Haj by Leon Uris?

The answer is incorrect (it is divided into three parts, ChatGPT claims). This is interesting. The Haj does not have a pdf version, so it is not part of the data set ChatGPT was trained on.

I asked ChatGPT:

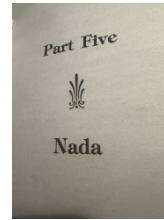
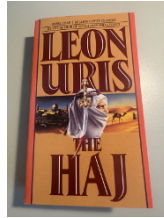
Be an expert on 20th century novels. How many parts are in The Haj by Leon Uris?

And the answer is:

As of my last update in April 2023, I don't have the specific titles of the parts in 'The Haj' by Leon Uris. For the most accurate and detailed information, including the titles of the parts, I recommend referring to a physical copy.

I did:

¹The creator of T_EX.



C.4 Asking good questions

There are several key considerations for asking effective questions (if you want to charge a lot, call this ‘prompt engineering’).

1. Specify the field of inquiry. For instance, as demonstrated in Section C.3, we initiated the question with

Be an expert on 20th century novels.

Since our focus is solely on LaTeX, you should begin with

Be an expert on LaTeX.

Typing just LaTeX is sufficient. Even better, once you instruct LaTeX in this way, it will remember the command.

2. Asking ChatGPT a question is just the beginning of a conversation. For example, ask:

Define \bmatrix

and after the long answer, respond

more examples

ChatGPT will then provide seven additional examples.

Next, respond with:

All examples should be at least 4 x 4 matrices.

and now all examples are 4×4 !

3. Let's think step by step. This is really amazing.

Ask: *Define \bmatrix, let's think step by step*

will teach you the use of the `\bmatrix` command in seven steps, so easy to understand.

To show you the responses would require several pages. Type these commands and see ChatGP's responses.

The next section introduces the The GPT Store containing Ai PDF (which handles PDF documents of up to 2GB PER FILE, uploads 1000s of PDF on myaidrive.com for free; it eliminates the need for repeated file uploads). Ai PDF helps make your questions more effective.

C.5 The GPT Store

In the second week of January, 2024, Open AI unveiled the GPT Store

<https://chat.openai.com/gpts>

As Reuters reported: “it is OpenAI’s attempt to build on the consumer success of ChatGPT, which introduced the world to generative AI last year, dazzling users with its ability to write humanlike prose and poetry.” We call the items in the store Special Purpose GPTs (GPT SPs).

The store also has top picks in seven categories; we select only two categories to discuss, Writing and Programming.

C.6 Style and grammar

In the Writing category, you find 12 items, most of them help you write better.

Here is a typical editing dialogue. We ask for help:

Edit for grammar and style, list changes by sentence:

Observe the prompt asks ‘list changes by sentence’. This is very important.

A typical English grammar book ranges from 400 to 1860 pages in length (the runner up is 1779).

These are general books not particularly useful for math. I think few mathematicians would want to spend time reading any of them. So this book is not trying to teach grammar, but it does utilize some terminology.

What we aim to do is simple and modest: select a few easy topics for discussion. If you follow our recommendations, you will eliminate 80% of your errors, making your papers easier to read and understand.

Note: 80%! Achieving 100% would require 600 pages!

I suggest reading a topic a day. It’s just a few minutes of easy reading.

Have fun.

Now the reply. It starts with a general appraisal of your prose.

Your writing has a conversational and engaging tone, which is excellent for drawing in readers. You’ve effectively communicated the purpose of the book and set a relaxed, approachable tone. Here are some suggestions to enhance the clarity and flow:

Then a sentence by sentence analysis:

Introduction: The opening sentence about English grammar books seems a bit out of context. Consider rephrasing to directly link it to the subject of your book. For example: “Unlike typical English grammar books, which can be extensive and overwhelming, this book adopts a more focused approach.”

Purpose and Audience: Clarify early on that the book is for mathematicians, to immediately establish relevance for your target audience.

Encouraging Tone: The part about eliminating 80% of errors is great for motivation. Consider adding a brief explanation of why 100% perfection isn't necessary, to further ease the pressure on the reader.

Call to Action: The suggestion to read a topic a day is excellent. You might want to add a line about the benefits of consistent, daily learning.

Closing: The closing lines are friendly and encouraging. You might want to add a final sentence that ties back to the goal of the book.

And finally, the revised version:

Stepping away from the daunting lengths of typical English grammar books, some spanning from 400 to 1860 pages, this guide presents a concise and mathematician-friendly approach. Our goal isn't to overwhelm you with extensive grammar rules, but to offer you key insights that are directly relevant and easily applicable.

Designed specifically for mathematicians, this book bypasses the complex jargon of standard grammar guides. We aim to simplify the process: selecting a handful of essential topics, ranging from [specific example] to [specific example]. By following our straightforward recommendations, you can reduce up to 80% of common errors in your papers, enhancing their clarity and readability. And don't worry about achieving 100% perfection – that would require an entire 600-page manual!

We suggest a manageable strategy: one topic a day. It's just a few minutes of easy reading that can significantly improve your writing skills.

Have fun and enjoy this journey towards clearer, more effective mathematical writing.

Much improved! This is really ChatGPT on steroids.

C.7 *T_EX programming*

In the Programming category, there are only three GPT SPs that do programming: Professional Coder (a gpt expert at solving programming problems), Auto Expert (your steadfast pair programmer, armed with enhanced code generation ability), and Code Guru (reviews code, writes pull requests, generates and optimizes functions, writes tests, and comments existing code).

We ask them for help to code a command `\threejump`, inspired by the command `\cleardoublepage`:

Create a LaTeX command named `\threejump` that will add `\newpage` commands until the page number is divisible by 3.

Professional Coder

```
\makeatletter
```

```

\newcommand{\threejump}{%
  \loop
    \ifnum\numexpr\value{page}%3\relax>0\relax
      \newpage
    \repeat
}
\makeatother

```

The relevant part of the log file:

```

(./profcoder.aux) [1{/usr/local/texlive/2023
/texmf-var/fonts/map/pdftex/updmap/
pdftex.map}]
./profcoder.tex:23: Undefined control sequence.
<inserted text> \TE@neg \intbool
                                     {\value {page}}{3}

1.23 \threejump

```

Auto Expert

```

\newcommand{\threejump}{
  \newcount\tempcount
  \tempcount=\value{page}
  \loop
    \ifnum\numexpr(\tempcount+2)/3*3\relax=\tempcount
      \else
        \newpage
        \advance\tempcount by 1
      \repeat
}

```

Code Guru

```

\usepackage{ifthen} % Required for if-then-else statements

\newcommand{\threejump}{
  \whiledo{\not\isodd{\thepage}}{ % Check if the page number
    is not divisible by 3
    \newpage
    \whiledo{\isodd{\thepage}}{ % If it is not, add a new page
      and check again
      \newpage
    }
  }
}

```

```
}
}
```

The relevant part of the log file:

```
./CodeGuru.tex:17: Undefined control sequence.
<inserted text> ...eg \(\value {tempcounter} \mod
                                                    3 = 0\))
1.17 \threejump
                                % Call your custom command here
?
```

C.7.1 Testing

As indicated by the log files, unfortunately, none of these three codes compile. Chulk this up to early days. When they will, this will open up a really easy way to be exposed to T_EX programming.

C.8 Tokens, darn it

Life would be easier without tokens. But we need them: ChatGPT uses them to measure the size of your prompts and answers and restricts them to 4096 tokens for ChatGPT 3.5 and twice as many for ChatGPT 4.0. Note that 4096 tokens is roughly equivalent to 6 to 8 pages of text.

To live within the token limits, draft clear and concise prompts. Write specific prompts to maximize your ChatGPT token allowance, especially if you want a more detailed response. Help ChatGPT avoid generating excessively long responses. For example,

What are the top five major tools to solve differential equations? Be concise and use bullet points.

Sometimes, you may need to provide specific details or elaborate on a query, which can use up more of your tokens. For instance, if you need to summarize a long meeting transcript, consider copying only the essential parts of the discussion. Use this prompt:

Summarize the main decisions in this conversation.

C.9 Conclusions

ChatGPT and the GPT SPs deserve deep study. Amazon's Kindle Store lists hundreds of books, including ChatGPT MASTERY, ChatGPT for Dummies, and The ChatGPT Millionaire. Even 'prompt engineering' has hundreds of listings. Udemy has thousands of ChatGPT video courses; prompt engineering has hundreds.

Luckily for us, by focusing on \LaTeX , our needs are limited. For instance, the Udemy course, ChatGPT Complete Guide, has 17 chapters on prompt engineering, and only a tiny chapter relevant to us.

I hope I succeeded in getting you started.

For a different take on the subject, see my article [\[24\]](#).

In digital streams,
Wisdom flows through ChatGPT,
Guiding thoughts to light.

Future bright and clear,
Free from illusions' shadow,
Hope's horizon near.

Bibliography

- [1] American Mathematical Society, User's Guide for the `amsmath` Package (version 2.10). Providence, RI, 2020.
- [2] American Mathematical Society, Using the `amsthm` package (version 2.20.3). Providence, RI, 2017.
- [3] Johannes Braams, Babel, a multilingual package for use with \LaTeX 's standard document classes. 2005, on CTAN.
- [4] Robert Bringhurst, The Elements of Typographic Style. Hartley & Marks Publishers, 2004.
- [5] Pehong Chen and Michael A. Harrison, Index preparation and processing. *Software Practice and Experience* **19** (9) (1988), 897–915.
- [6] The Chicago Manual of Style. 17th edition. University of Chicago Press, Chicago, 2017.
- [7] Jacques Crémer, A very minimal introduction to TikZ.
<https://cremeronline.com/LaTeX/minimaltikz.pdf>
- [8] Michael Doob, \TeX Starting from 1. Springer-Verlag, New York, 1993.
- [9] Lyn Dupré, BUGS in Writing. A Guide to Debugging Your Prose. 2nd edition. Addison-Wesley Professional, Reading, MA, 1998.
- [10] H. W. Fowler, A Dictionary of Modern English Usage. Oxford University Press, 2023.
- [11] Enrico Gregorio, Horrors in \LaTeX : How to misuse \LaTeX and make a copy editor unhappy. *TUGboat* **26** (2005), 273–279.
- [12] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach, The \LaTeX Graphics Companion. Addison-Wesley, Reading, MA, 1997.
- [13] Michel Goossens and Sebastian Rahtz (with Eitan Gurari, Ross Moore, and Robert Sutor), The \LaTeX Web Companion: Integrating \TeX , HTML and XML. Addison-Wesley, Reading, MA, 1999.
- [14] George Grätzer, Math into \TeX : A Simple Introduction to AMS- \LaTeX . Birkhäuser Boston, 1993.

- [15] George Grätzer, AMS- \LaTeX . Notices Amer. Math. Soc. **40** (1993), 148–150.
- [16] George Grätzer, Advances in \TeX . IV. Header and footer control in \LaTeX . Notices Amer. Math. Soc. **41** (1994), 772–777.
- [17] George Grätzer, General Lattice Theory. 2nd edition. Birkhäuser Verlag, Basel, 1998.
- [18] George Grätzer, Math into \LaTeX . 3rd edition. Birkhäuser Verlag, Boston, Springer-Verlag, New York, 2000.
- [19] George Grätzer, More Math into \LaTeX . 4th edition. Springer-Verlag, New York, 2007.
- [20] George Grätzer, More Math into \LaTeX . 5th edition. Springer-Verlag, New York, 2016.
- [21] George Grätzer, The Congruences of a Finite Lattice, A Proof-by-Picture Approach, third edition. Birkhäuser, 2023.
- [22] George Grätzer, Turbulent transition, *TUGboat* **21** (2001), 111–113.
- [23] G. Grätzer, The gods smile at me: *The \LaTeX Companion*, third edition, and ChatGPT. *TUGboat* **44** (2023), 319–321.
- [24] G. Grätzer, ChatGPT 101 for \LaTeX users. EMS Magazine, September 2024 issue.
- [25] Harley Hahn, Harley Hahn’s Student’s Guide to Unix. 2nd edition. McGraw-Hill, New York, 1993.
- [26] Horace Hart, Hart’s Rules For Compositors and Readers at the University Press, Oxford. Oxford University Press, Oxford, 1991.
- [27] How to draw a regular pentagon with compass and straightedge.
[https://math.stackexchange.com/questions/2102517/
how-to-draw-a-regular-pentagon-with-compass-and-straightedge](https://math.stackexchange.com/questions/2102517/how-to-draw-a-regular-pentagon-with-compass-and-straightedge).
- [28] Uwe Kern, Extending \LaTeX ’s color facilities: the `xcolor` package. 2005.
- [29] Donald E. Knuth, The Art of Computer Programming. Volumes 1–3. Addison-Wesley, Reading, MA, 1968–1998.
- [30] Donald E. Knuth, The \TeX book. Computers and Typesetting. Vol. A. Addison-Wesley, Reading, MA, 1990.
- [31] Donald E. Knuth,
<https://www-cs-faculty.stanford.edu/~knuth/chatGPT20.txt>
- [32] Leslie Lamport, \LaTeX : A Document Preparation System. 2nd edition. Addison-Wesley, Reading, MA, 1994.
- [33] Philipp Lehman, The Bib \LaTeX Package.
- [34] Ruari McLean, The Thames and Hudson Manual of Typography. Thames and Hudson, London, 1980.
- [35] Frank Mittelbach with Ulrike Fischer, The \LaTeX Companion. Third edition. Parts I and II. Addison-Wesley, 2023. Referenced as LC3.

- [36] Oren Patashnik, `BIBTeX`ing. Document in the `BIBTeX` distribution.
- [37] Oren Patashnik, `BIBTeX` 1.0. *TUGboat* **15** (1994), 269–273.
- [38] Robert M. Ritter, *New Hart’s Rules: The Handbook of Style for Writers and Editors*, Oxford. Oxford University Press, Oxford, 2005.
- [39] Rainer Schöpf, A new implementation of the `LaTeX` `verbatim` and `verbatim*` environments. *TUGboat* **11** (1990), 284–296.
- [40] Raymond Seroul and Silvio Levy, *A Beginner’s Book of TeX*. Springer-Verlag New York, 1995.
- [41] Straightedge and compass construction.
https://en.wikipedia.org/wiki/Straightedge_and_compass_construction.
- [42] Ellen Swanson, *Mathematics into Type*. Updated edition. Updated by Arlene Ann O’Sean and Antoinette Tingley Schleyer. American Mathematical Society, Providence, RI, 1999.
- [43] Till Tantau, *User’s Guide to the Beamer Class*. 2005.
- [44] Till Tantau, editor, *The TikZ and PGF Packages*. <https://tikz.dev>
- [45] Gérard Tisseau and Jacques Duma, *TikZ pour l’impatient*, 2015.

Index

Italic numbers indicate figures or tables, *bold* numbers indicate definitions.

- `\` (backslash), 163, 548
 - key, 7, 63
 - starts commands, 6, 11, 54
 - text symbol, 63, 67, 539
- `_` (space), 9, 49, 50
 - and `\verb*` command, 128
 - in arguments of commands, 72
 - in `\bibitem` labels, 273
 - in `\cite` commands, 273
 - in command names, 55
 - in tabular environments, 115
 - text symbol, 549
- `_` (space com.), 9, 52, 57, 81, 190, 550
- `!` (exclamation mark), 7, 48
 - float control, 267, 529
 - in `\index` commands, 503, 504, 507
- `¡` (exclamation mark, Spanish), 67, 545, 548
- `\!` (negthinspace), 181, 190, 544, 550
- `\"` (" dieresis/umlaut text accent), 7, 11, 66, 145, 546
- `"` (double quote), 7, 61
 - in BibTeX database fields, 489
 - in `\index` commands, 507
 - key, 7, 49, 61
- `"ck` (European character), 145
- `"s` (eszett), 145
- `#`
 - in custom commands, 411, 420
 - key, 7, 49
- `\#` (# octothorp), 63, 67, 548
- `$`
 - as inline math delimiter, 12, 13, 58, 60, 150
 - act as braces, 150
 - must be balanced, 151
 - in error messages, 39, 40, 128, 151, 152, 154, 238
 - key, 7, 49
- `\$` (\$ dollar sign), 7, 63, 67, 548
- `$$`
 - in error messages, 39
 - TeX displayed math delimiter, 43, 150
- `%`
 - as comment character, 47, 71–73, 121, 303, 508
 - in BibTeX databases, 72, 487
 - key, 49
- `\%` (% percent), 7, 47, 63, 67, 73, 548
- in e-mail addresses, 286
- `&`
 - as alignment point, 21, 23, 220, 222, 224, 225
 - as column separator, 17, 115, 224, 225, 236, 241
 - key, 7, 49
- `\&` (& ampersand), 63, 67, 548
- `'` (right single quote), 7, 9, 48, 61, 549
 - for primes ('), 15, 156, 198
- `\'` (´ acute text accent), 66, 546
- `\(` (start inline math mode), 13, 60
 - acts as special brace, 150, 151
 - must be balanced, 151
- `(`
 - as math delimiter ((), 163, 539
 - in index entries, 504
 - key, 7, 48
- `\)` (end inline math mode), 13, 60

- acts as special brace, 150, 151
- must be balanced, 151
-)
 - as math delimiter (\rangle), 163, 539
 - in index entries, 504
 - key, 7, 48
- * (* asterisk), 63
 - interactive prompt, 94, 418–419
 - key, 7, 49
- *-form
 - of commands, 55, 77, 80, 83, 85, 109, 110, 128, 198, 200, 247, 257, 318, 413, 416, 423, 426, 483, 517, 519
 - of environments, 154, 205, 215, 217, 219, 235, 265
- + key, 7, 49
- + (plus), 15, 49, 155, 188, 191, 192, 217, 220–222
 - as binary operation, 536
- + and – rule, 188, 191, 192, 222
 - and subformulas, 220
 - in `multline` environments, 217
- \, (thinspace), 16, 53, 61, 190, 544, 550
- , (comma), 7, 48
 - and italic correction, 137
 - in `\bibitem` commands, 273
 - in BibTeX databases, 468, 469, 488
 - use in formulas, 153
- (dash, hyphen, minus), 7, 15, 49, 61, 62, 155, 188, 191, 192, 199, 220, 221
 - as binary operation, 536
 - in `\hyphenation` commands, 68
- \- (opt. hyphen), 12, 68, 76
- (– number ranges, en dash), 11, 62, 67, 548
 - in BibTeX databases, 471
- (— em dash), 11, 62, 67, 548
- . (period), 7, 48
 - and italic correction, 137
 - in BibTeX databases, 470
- \. overdot text accent ($\dot{}$), 66, 546
- \/ (italic correction), 137–138, 550
 - and font commands, 138
- / (slash)
 - as divisor, 15, 49, 155
 - as math delimiter ($/$), 163, 539
 - key, 7
- \: (medspace spacing com.), 190, 544, 550
- : (colon), 7, 48
 - as binary relation, 43, 49, 191, 533
- \; (thickspace), 190, 544, 550
- ; (semicolon), 7, 48
- < (less than)
 - as binary relation ($<$), 533
 - key, 12, 49
 - text symbol, 67, 548
- \=
 - macron text accent (̄), 66, 546
 - tab set command, 121
- = (equal sign)
 - as binary relation ($=$), 533
 - in BibTeX database fields, 468
 - key, 7, 49
- \> (tabbing com.), 121, 123
- > (greater than)
 - as binary relation ($>$), 533
 - key, 12, 49
 - text symbol, 67, 548
- ?
 - question mark, 7, 48
- ¿ (Spanish question mark), 67, 545, 548
- @ (@ at sign), 7, 49, 63
 - in BibTeX database (bib) files, 468
 - in `\index` commands, 505, 506
- @-expression, 116
- ⌀. (blank math symbol for comm. diagrams), 245, 246
- \@. (intersentence space), 53, 550
- @<<< (stretchable left arrow math sym.), 245
- @= (stretchable equals sign math sym.), 245
- @>>> (stretchable right arrow math sym.), 245
- @AAA (stretchable up arrow math sym.), 245
- @VVV (stretchable down arrow math sym.), 245
- @\vert (stretchable double vertical line), 246
- \[(start displayed math mode), 14, 19, 43, 150, 153
 - acts as special brace, 150, 154
- [
 - and optional arguments, 18, 55, 56, 108, 112, 115
 - key, 7, 48
 - math delimiter ($[$), 163, 539
 - with `\item` commands, 102
 - with `\newcommand`, 410
- \] (end displayed math mode), 14, 19, 43, 150, 153
 - acts as special brace, 150, 154
-]
 - and optional arguments, 18, 55, 56, 108, 112, 115
 - key, 7, 48
 - math delimiter ($]$), 163, 539
 - with `\item` commands, 102
 - with `\newcommand`, 410
- {
 - for required arguments, 6, 11, 14, 55, 56, 156, 406
 - for scoping, 58–59
 - in BibTeX entries, 468, 472
 - key, 7, 49
 - must be balanced, 38, 41, 58, 64

- in `\index` commands, 512
 - with `\def`, 419
- `\{`
 - as math delimiter (`{`), 43, 163, 539
 - text brace (`{`), 63, 67, 548
- `}`
 - for required arguments, 6, 11, 14, 55, 56, 156, 406
 - for scoping, 58–59
 - in \LaTeX entries, 468, 472
 - key, 7, 49
 - must be balanced, 38, 41, 58, 64
 - in `\index` commands, 512
 - with `\def`, 419
- `\}`
 - as math delimiter (`}`), 43, 163, 539
 - text brace (`}`), 63, 67, 548
- `^` (caret)
 - for superscripts, 15, 16, 156–157, 159, 180
 - key, 7, 49, 63
- `\^` circumflex text accent (`^`), 66, 67, 546, 548
- `_` (`_` underscore), 7, 63, 67, 548
 - in e-mail addresses, 286
- `_` (underscore)
 - for subscripts, 15, 16, 156–157, 159, 174, 180
 - key, 7, 49
- `‘` (left single quote), 7, 9, 48, 61, 549
- `\`` (``` grave text accent), 48, 66, 546
- `~` (tilde)
 - key, 7, 9, 49
 - tie/unbreakable space, 9, 49, 52, 62, 78, 550
 - absorbs spaces, 63
 - in \LaTeX databases, 471
 - with cross-references, 262
- `\~` (`~` tilde text accent), 63, 66, 546, 549
- `9pt` (doc. class option), 306
- `10pt` (doc. class option), 84, 306
- `11pt` (doc. class option), 306
- `12pt` (doc. class option), 138, 306
- `6colorpages` (sample file), xxvi, 327, 329, 330, 345, 399
- `\\` (new line), 76, 78
 - and `\kill` commands, 123
 - breaking lines with, 17, 21, 23, 25, 77, 104, 115, 121, 123, 125, 172, 215, 216, 219, 220, 231, 235, 247, 281–285, 306, 353
 - in arguments of commands, 78
 - in environments, 78, 104, 125
 - in text, 78
 - optional argument of, 76–78, 247
- `*` (new line), 77, 247
- `a4paper` (doc. class opt.), 307
- `\AA` (\AA), 67, 545
- `\aa` (\aa), 67, 545
- abbreviations
 - in bibliographic entries, 480–481
 - defining, 480
 - periods in, 52–53
 - using small caps for, 136, 524
 - using ties (`~`) with, 52
- absolute
 - names for equations, 21
 - referencing, 21, 99, 206, 263
 - units, 12, 76, 85, 87, 92, 134, 138, 142, 447
- abstract (bibl. com.), 469
- abstract (text env.), 26, 54, 255, 292
 - in `beamer` document class, 322
 - placement of, 255, 292
- `\abstractname` (redef. name), 416
- abstracts, 26, 255, 322
 - in AMS document classes, 292
 - separate page for, 253, 308
- accents
 - European, 11, 65–66, 549
 - hyphenation of words with, 68
 - in bibliographies, 472
 - in `hyperref` titles, 317
 - math, 17, 173, 182, 183, 310, 418, 526
 - double, 173
 - text, 7, 11, 55, 65–66, 546, 549
- accents (pack.), 173
- acronyms, using small caps for, 136, 524
- acute (`´` acute text accent), 66, 546
- `\acute` (\acute{x} math accent), 172, 543
- `\addcontentsline` (table of contents com.)
 - and lists of figures and tables, 522
 - arguments of, 520, 522
- adding lines
 - to lists of figures and tables, 522
 - to tables of contents, 519–520, 530
- addition, 15, 49, 155
- `\address` (top matter com.), 305, 322
 - optional arguments of, 284, 285
 - rules for using, 284–285
 - with multiple authors, 288
- address (\LaTeX database field), 468, 469
- addresses
 - of authors in articles, 284, 288, 305
- `\addtocontents` (table of contents com.)
 - and lists of figures and tables, 522
 - arguments of, 520, 522
- `\addtocounter` (counter incr. com.), 446
- `\addtolength` (counter setting com.), 449
- adjusted
 - columns, 214, 236–247
 - formulas, 214, 236–247
 - flush left and right, 218

- adjusting
 - interline spacing, 77–142
 - placement of root with `\sqrt`, 160
- Adobe
 - Acrobat Professional, 317, 321, 526
 - placing hyperlinks in documents, 315
 - Illustrator, 28
 - Reader, 28, 317, 321, 353, 529
- `\AE` Aesc (Æ), 67, 545
- `\ae` aesc (æ), 67, 545
- afterwords, of books, 517
- `\aleph` (ℵ Hebrew char.), 86, 162, 531
- `\alert` (beamer text style com.), 334
- `align` (math align. env.), 21–24, 43, 212–214, 223–228, 234, 235, 526
 - annotated, 213
 - page breaks in, 247
- `align*` (math align. env.), 23, 225, 231
- `alignat` (math align. env.), 43, 212, 214, 228–230
 - argument of, 230
- aligned
 - formulas, 21–24, 211–214, 223–230, 234, 236–237, 247, 307
 - text in, 230–231
 - math environments, 247
 - `align`, 21–24, 43, 212, 214, 223–228, 234, 235, 247
 - `alignat`, 43, 212, 214, 228–230
 - and `\verb` commands, 128
 - `eqnarray`, 226–227
 - `flalign`, 212, 214, 225, 235
 - inserting lines of text in, 230
 - intercolumn space in, 211
 - subsidiary math environments, 231–236
 - `aligned`, 214, 232–234
 - `alignedat`, 232–234, 244
 - `split`, 212, 214, 234–236, 307
- `aligned` (subsid. math align. env.), 214, 232–234
 - and `\allowdisplaybreaks` commands, 248
- `alignedat` (subsid. math align. env.), 232–234, 244
 - and `\allowdisplaybreaks` commands, 248
- alignment
 - of columns in tabular environments, 115, 116
 - of formulas, *see also under* adjusted, 223–236
 - annotated, 24, 213
 - commands for, 218
 - flush left and right, 211
 - multiline, 21–25, 221–223
 - simple, 21–24
 - of large symbols, 233–234
 - of text, 54, 86, 104, 114, 231, 451
 - centering, 7, 10, 43, 104, 105, 114, 117, 231, 459
 - command declarations for, 104, 117
 - flush left, 10, 104
 - flush right, 10, 86, 104
 - with `trivlist` environments, 459
 - of text boxes, vertical, 92
 - point for subformulas (&), 21, 23, 25, 220, 222, 225
 - within text boxes, 87, 92
- `allowdisplaybreaks` (display break com.), 247
 - and subsidiary math environments, 248
 - optional argument of, 248
- `alltt` (disp. text env.), 157
- `alltt` (pack.), 127
- `\Alph` (uppercase letter counter style com.), 445
- `\alph` (lowercase letter counter style com.), 445
- `\alpha` (α Greek char.), 162, 532
- alpha (BIBTeX style), 472
- alphabets (counter styles), 445
- alphabets, math, 199–200
 - blackboard bold, 180, 201, 433, 543
 - calligraphic, 184, 200, 310, 433, 543
 - Euler Fraktur, 179, 181, 201, 310, 433, 543
 - Euler Script, 310, 311, 433, 543
 - Greek, 200
 - symbol, 200–202
- alphanumeric keys, 7
- `\amalg` (II math op.), 536
- American Mathematical Society, *see under* AMS, xxi, 3
- ampersand (&)
 - as alignment point
 - for annotations, 23, 25
 - for subformulas, 21, 23, 220, 222, 225
 - as column separator, 17, 115, 224, 225
 - text symbol, 63, 67, 548
- AMS, 68, 93, 96, 143, 247, 258, 273, 279–282, 287, 288, 311, 321, 415, 490, 491, 517, 519
- AMSFonTS, *see* AMSFonTS
- article templates, 303–306
- bibliographies
 - database files, 481
 - fields, 469
 - sample files, 481
 - styles, 273, 465, 471, 481
- distribution, 279, 309–311, 421, 480, 481
- document classes, *see* document classes
- environments, *see* displayed math environments, subsidiary math environments, and text environments
- subject classifications, 287
- top matter

- AMS specific information, 287–288
- article information, 281–283
- author information, 283–287
- commands, 281–292, 305
- examples of, 289–292
- rules for, 283–287
- with multiple authors, 288–289
- Web site, 288
- `\AmS` ($\mathcal{A}\mathcal{M}\mathcal{S}$ logo com.), 66
- AMS package distribution, 279, 309–311, 321, 421, 461, 480, 481
- $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$
 - version 1.1
 - two-letter font commands, 140
- `amsalpha` ($\text{\BIB}\text{\TeX}$ style), 472
- `amsart` (doc. class), 34, 42, 68, 105, 151, 193, 253–255, 258, 279–311, 515, 518
 - anatomy of, 25
 - and numbered lists, 98
 - front matter of, 292
 - sample article, 105, 109, 111, 234, 274, 303, 436–465
 - top matter of, 255
- `amsart.tpl` (sample file), 303
- `amsbook` (doc. class), *xxxi*, 34, 257, 417, 515–518
 - for exercises, 522
- `amsbsy` (pack.), 310, 311
- `amscd` (pack.), 244–247, 310
- AMSFonTS, 309–311
 - document class options for, 309
- `amsfonts` (pack.), 310–311
- `amsgen` (pack.), 310, 311
- `amsmath` (pack.), 34, 236, 309, 311, 447
 - document class options affecting, 307–308
 - options, 308
- `amsmath.sty` (AMS distr. file), 428
- `amsmidx` (pack.), 311
- `\AMSname` (`hyperref` redef. name), 319
- `amsopn` (pack.), 310, 311
- `amspain` ($\text{\BIB}\text{\TeX}$ style), 465, 481, 483
- `amspain.bst` ($\text{\BIB}\text{\TeX}$ style file), 465, 471, 481, 483, 489
- `amsproc` (doc. class), 281, 417
- `amsproc template` (AMS sample template file), 306
- `amsrefs` 2002, 491
- `amssymb` (pack.), 178, 303, 309, 310, 415
- `amstext` (pack.), 310, 311
- `amsthm` (pack.), 34, 311
- `amsxtra` (pack.), 173, 309, 310
- `\And` ($\&$ math op.), 536
- and, in bibliographies, 470
- `\angle` (\angle math sym.), 538
- annotations
 - alignment of, 23, 24
 - of formulas, 229
- `AnyStyle`, 498
- `apalike` (pack.), 481
- apostrophe (`'`), 9, 48
 - key, 15, 198
- appendices, 259, 517
 - numbering of, 259, 517
- `\appendix` (struct. com.), 259, 517
- `\appendixname` (redef. name), 417
 - and `hyperref` package, 319
- `\approx` (\approx binary rel.), 533
- `\approxeq` (\cong binary rel.), 534
- `\arabic` (numeric counter style com.), 445, 455
- `\arccos` (\arccos math op.), 168, 540
- `\arcsin` (\arcsin math op.), 168, 540
- `\arctan` (\arctan math op.), 168, 540
- `\arg` (\arg math op.), 168, 540
- arguments
 - empty (`{ }`), 57, 156, 175, 188, 189, 197, 208, 287
 - long, 60
 - movable, 60, 521
 - negative, 82, 160
 - of commands, *see also under* specific commands, *see also under* specific commands, 323, 331, 344, 346, 350
 - of commutative diagram symbols, 245
 - of custom commands, 410–413
 - multiple, 410, 412
 - optional, 414
 - short, 413–414
 - of custom environments
 - optional, 421, 424–426
 - short, 426
 - of environments, *see also under* specific environments
 - of top matter commands, *see also* top matter, 280, 281, 287–289, 292
 - optional, 57, 78, 80, 93, 94, 96, 106, 159, 160, 175, 238, 247, 254
 - multiple, 56
 - of commands, 100, 102, 103, 107, 203, 205, 273, 284–287, 343, 444, 446, 451, 453, 454, 458, 522
 - of environments, 56, 106, 108, 112, 115, 233, 248, 267, 425, 529
 - of sectioning commands, 257, 519
 - of structural commands, 259, 516
 - of top matter commands, 281–289
 - use square brackets (`[]`), 18, 55–56, 103, 112, 115, 292
 - required, 92, 93, 175, 197, 280, 281
 - multiple, 14, 56, 117, 194, 196, 197, 277, 420, 448–450

- of commands, 14, 55, 117, 194, 196, 197, 261, 277, 287, 448, 520, 546
 - of environments, 56, 115, 116, 228, 230, 241, 242, 271–273, 425–427, 522
 - use braces (`{ }`), 6, 11, 14, 55, 56
 - short, 60
 - single character, 56
 - spacing within, 14, 72, 87, 152, 160, 179, 194, 196, 511
 - specifying zero in, 95
- arithmetic
 - operations, 15, 155–157
 - with counters, 446–447
 - with length commands, 449
- array (subsid. math env.), 212, 214, 233, 237, 241–244
 - arguments of, 241, 242
- arrays, 17–18, 241–243
- `\arraystretch` (table com.), 119
 - adjusting vertical spacing with, 119
- arrows, math, 175, 187, 245, 537
 - as delimiters, 163, 164, 539
- `arrows.meta`, TikZ library, 362
- `art1.tex` (sample file), 25, 37, 38, 280, 322, 324
- `art1ill.tex` (sample file), 27
- `art1ind.idx` (index entry file), 509–511
- `art1ind.ilg` (index log file), 511
- `art1ind.ind` (index proc. file), 509, 511
- `art1ind.tex` (sample file), 502, 510, 511
- `art2-ref.tex` (sample file), 316, 319
- `art2.tex` (sample file), 105, 109, 111, 234, 274, 280, 281, 293–299, 412, 436, 481
 - source file, 293–299
 - typeset, 293–302
- `art2biblatex.tex` (sample file), 492
- `art2bibtex.aux` (sample aux. file), 485
- `art2bibtex.blg` (sample BIB_T_E_X log file), 486
- `art2bibtex.tex` (sample file), 465, 481, 483, 484, 487
- `art2ccom.tex` (sample file), 436
- `art3.tex` (sample file), 492, 495
- ARTICLE (bibl. entry type), 468, 472
- article (doc. class), 26, 34, 193, 253, 255, 311–417
 - sectioning commands provided by, 258
- articles
 - bibliographies in, 274, 467, 480–489
 - creating templates for, 303–306
 - in BIB_T_E_X database files, 472–473, 480
 - sectioning of, 255–259
 - top matter information
 - AMS specific, 281–292
 - AMS subject classifications, 287
 - author addresses, 284, 288, 305
 - author names, 283, 305
 - contributor, 284
 - current addresses, 285, 289
 - dedications, 282
 - e-mail addresses, 285, 288, 305
 - Internet addresses, 286, 288, 305
 - keywords, 288, 306
 - research support, 287, 288, 305
 - title, 281, 306
- `\ast` (`*` math op.), 536
- asterisk (`*` text symbol), 67, 548
- `\asymp` (\asymp binary rel.), 533
- at sign (`@`)
 - in BIB_T_E_X database (bib) files, 468
 - in `\index` commands, 505, 506
- `\author` (top matter com.), 253, 312, 327, 518
 - optional arguments of, 283, 327
- author (BIB_T_E_X database field), 468
- authors
 - information about
 - in AMS top matter, 283–288
 - multiple
 - in AMS documents, 288
 - in bibliographies, 470
 - names of
 - in articles, 283, 305
 - in running heads, 283
- Auto Expert, 556
- automatic
 - numbering, 19, 21, 36, 99, 106, 235, 256
 - renumbering, 20, 36
- `\autoref` (hyperref cross-ref. com.), 316, 318
 - names supported by, 319
- aux (aux. files), 36, 484, 521, 530
- auxiliary files, 36, 522, 529, *see also* aux, bbl, bib, blg, bst, glo, idx, ilg, ind, lof, log, lot, out, and toc
 - polishing, 529–530
- `\b` (`_` underscore text accent), 66, 546
- Bezier cubics
 - draw (`\draw Bézier` cubics, 367
- babel (pack.), 145–146
 - options, 146
- `babybeamer1.pdf` (sample file), 327–332
- `babybeamer2.tex` (sample file), 330, 334, 336
- `babybeamer3.tex` (sample file), 336
- `babybeamer4.tex` (sample file), 337–339
- `babybeamer4block.tex` (sample file), 338
- `babybeamer5.tex` (sample file), 339–342
- `babybeamer6.tex` (sample file), 342
- `babybeamer7.tex` (sample file), 344
- `babybeamer8.tex` (sample file), 354
- back matter, 253, 268, 275, 517
 - bibliographies in, 253

- index in, 500
 - numbering of chapters in, 517
- backend for Bib \LaTeX , 492
- `\backepsilon` (\ni binary rel.), 534
- `\backmatter` (struct. com.), 517
- `\backprime` (\backprime math sym.), 538
- `backref` (opt. of `hyperref` pack.), 317
- `backref` (pack.), 316, 317
- `\backsim` (\sim binary rel.), 534
- `\backsimeq` (\backsimeq binary rel.), 534
- `\backslash` (\backslash math delimiter), 163, 539
- `backslash` (\backslash text symbol), 67, 548
- balancing
 - braces, 58, 64
 - errors with, 58–59
 - in `\index` commands, 512
 - inline math delimiters, 151
 - math delimiters, 164, 221
- `\bar` (\bar{x} math accent), 17, 172, 173, 183, 543
- bar, vertical ($|$ text symbol), 67, 548
- `\barwedge` (\barwedge math sym.), 536
- baseline, of text, 88, 142
 - adjusting with `setspace`, 78
- `\baselineskip` (length com.), 80, 142
- `\baselinestretch` (length com.), 78
- `\Bbbk` (\mathbb{k} math sym.), 538
- `\BBigl` (math delim.), 166
- `\BBigr` (math delim.), 166
- `bbl` (proc. bibl. files), 489
- beamer (doc. class), *see also under* presentations, 321–355
 - abstracts in, 322
 - color in, 344
 - commands
 - `\beamergotobutton`, 339
 - `\breakhere`, 346
 - `\color`, 335, 344
 - `\colorbox`, 344
 - `\fcolorbox`, 344
 - `\frametitle`, 29–31, 323, 324, 339
 - `\hyperlink`, 339
 - `\institute`, 322
 - `\note`, 350
 - `\only`, 327, 330–332, 334–335, 338
 - `\onslide`, 327, 331, 332, 334–335, 338
 - `\pause`, 31, 82, 327, 330, 333
 - `\setbeamer`, 350
 - `\setbeamercolor`, 345
 - `\tableofcontents`, 324
 - `\textcolor`, 344
 - `\usecolortheme`, 352
 - `\usefonttheme`, 352
 - `\usetheme`, 28, 324, 351
 - documentation for, 322, 354
 - options
 - handout, 353
 - `notes=only`, 350
 - `notes=show`, 350
 - `trans`, 350, 353
 - `xcolor=dvipsnam`, 344
 - preamble in, 28, 322–323
 - sections in, 324
 - sidebars, 324
 - sidebars in, 324
 - table of contents in, 324
 - themes, 28, 324, 351–352
 - top matter in, 28, 322–323
- `\beamergotobutton` (beamer com.), 339
- `beamerstructure.tex` (sample file), 347, 349
- `\because` (\because binary rel.), 534
- Beeton, Barbara, 71
- `\begin` (start of env.), 54, 58, 59
- Berkeley (beamer theme), 324, 327, 328, 351
- `\beta` (β Greek char.), 162, 532
- `\beth` (\beth Hebrew char.), 162, 531
- `\between` (\between binary rel.), 534
- Bezos, Javier, 173
- `\bf` (obs. \LaTeX 2.09 font com.), 140
- `\bfseries` (font weight com. dec.), 58–60, 104, 135, 138–140, 546
- `bfseries` (font weight env.), 104
- `bib` (Bib \TeX database files), 465, 467–481, 483
 - accents in, 472
 - AMS, 480, 481
 - commas ($,$) in, 468, 469, 488
 - comments in, 487
 - cross-referencing within, 476–477
 - delimited with parentheses ($(\)$), 468
 - delimiters in, 468
 - double quote ($"$) in, 489
 - en dashes ($-$) in, 471
 - entries
 - abbreviations in, 480–481
 - capitalization in, 471
 - Jr. in, 470
 - multiple authors in, 470
 - von in, 470
 - entry types, 467–481
 - case-sensitivity of, 469
 - start with \emptyset , 468
 - equals signs ($=$) in, 468
 - fields in, 469, 475–478, 480
 - adding your own, 469
 - case-sensitivity of, 469
 - optional, 470, 472–480
 - required, 470, 472–480
 - rules for typing, 470–481
 - termination of, 489
 - used by AMS bibliography styles, 469
 - location of, 483

- number ranges in, 471
- numbers in, 468
- periods (.) in, 470
- portability of, 470
- samples of, 467, 481
- ties (unbreakable spaces) in, 471
- `biber`, 492, 496, 497
- `\bibitem` (bibl. com.), 20, 272
 - commas in, 273
 - labels for, 271
 - optional argument of, 273
 - spaces in, 273
- `BibLaTeX`, 492, 495, 496, 496
- `BibLaTeX`, 491, 492, 495
- bibliographies, 26, 268, 274
 - AMS support for, 480, 481
 - appear in back matter, 253
 - citing references from, 20, 273, 482
 - as hyperlinks in PDF files, 317
 - with `BIBTeX`, 482
 - commands for, 480, 482
 - defining, 53, 268–274, 415, 451, 467–481
 - examples of, 268–271, 465–467
 - in articles, 467, 480, 489
 - multiple, in a document, 274
 - numbering of, 273
 - portability of, 470, 472
 - sample files for, 481
 - styles, 465, 466, 481
 - AMS, 273, 465, 471, 481, 483
 - templates, 268
- `\bibliography` (bibl. com.), 56, 481, 483
- `\bibliographystyle` (bibl. com.), 481, 483
- `\bibname` (redef. name), 274, 417
- `Bibsonomy`, 497
- `BIBTeX`, 317, 465
 - AMS support for, 480
 - citing references with, 482
 - commands, 480, 482
 - log files, *see* `blg`
 - running, 481–489
 - sample files for, 481
 - style files, *see* `bst`
 - styles, 465, 466, 470, 481, 483
- `BIBTeX`, 491
- `BIBTeX1.0` (Patashnik), 561
- `Bibtex2html`, 497
- `bibtexcross.tex` (sample file), 496
- `BIBTeXing` (Patashnik), 490
- `BibTool`, 497
- `\Big` (math delim. size com.), 165
- `\big` (math delim. size com.), 165
- `\bigcap` (\bigcap large math op.), 171, 542
- `\bigcirc` (\bigcirc math op.), 536
- `\bigcup` (\bigcup large math op.), 171, 542
- `\Bigg` (math delim. size com.), 165
- `\bigg` (math delim. size com.), 165
- `\Biggl` (math delim. size com.), 165
- `\biggl` (math delim. size com.), 165, 183
- `\biggm` (math delim., as binary rel. size com.), 167
- `\Biggr` (math delim. size com.), 165
- `\biggr` (math delim. size com.), 165, 183
- `\Bigl` (math delim.), 166
- `\Bigl` (math delim. size com.), 165
- `\bigl` (math delim.), 166
- `\bigl` (math delim. size com.), 165
- `\bigm` (math delim., as binary rel. size com.), 167
- `\bigodot` (\bigodot large math op.), 171, 542
- `\bigoplus` (\bigoplus large math op.), 171, 542
- `\bigotimes` (\bigotimes large math op.), 171, 542
- `\Bigr` (math delim.), 166
- `\Bigr` (math delim. size com.), 165
- `\bigr` (math delim.), 166
- `\bigr` (math delim. size com.), 165
- `\bigskip` (spacing com.), 42, 84
- `\bigsqcup` (\bigsqcup large math op.), 171, 542
- `\bigstar` (\bigstar math sym.), 538
- `\bigtriangledown` (\bigtriangledown math op.), 536
- `\bigtriangleup` (\bigtriangleup math op.), 536
- `\biguplus` (\biguplus large math op.), 171, 542
- `\bigvee` (\bigvee large math op.), 171, 542
- `\bigwedge` (\bigwedge large math op.), 171, 542
- binary operations, 188, 198, 536
 - + and – rule, 191, 217, 220
 - adding white space around, 153
 - and alignment of displayed formulas, 222
 - and breaking displayed formulas, 217, 220, 221
 - created with `\overset`, 196
 - created with `\underset`, 181
 - defining, 198
 - and subformulas, 220
 - for congruences, 169
- binary relations, 43, 167, 180, 188, 189, 191, 533, 534
 - adding white space around, 153
 - and alignment of displayed formulas, 222
 - and breaking displayed formulas, 221
 - created with `\overset`, 181, 196
 - defining, 198
 - delimiters as, 167
 - math delimiters as, 167
 - negated, 180, 197, 535
- `\binom` (math com.), 15, 157
- binomial coefficients, 15, 157
 - in inline and displayed math environments, 157
- blackboard bold (math alphabet), 180, 201, 433

- `\blacklozenge` (◆ math sym.), 538
- `\blacksquare` (■ math sym.), 538
- `\blacktriangle` (▲ math sym.), 538
- `\blacktriangledown` (▼ math sym.), 538
- `\blacktriangleleft` (◀ binary rel.), 534
- `\blacktriangleright` (▶ binary rel.), 534
- blank
 - lines
 - in bibliographies, 317
 - in displayed math environments, 39, 151, 154, 219
 - in inline math environments, 151
 - in text environments, 43, 98
 - in top matter commands, 281
 - in `verbatim` environments, 126
 - terminating paragraphs with, 9, 43, 50, 79, 98, 124, 317
 - math delimiters, 164, 181
 - math symbol for commutative diagrams (⓪), 245
- `blg` (BIB_T_EX log files), 486
- `block` (beamer env.), 338
- blue spaces, *see also* tie, unbreakable spaces, non-breakable spaces, 62
- `Bmatrix` (subsid. math env.), 240
- `bmatrix` (subsid. math env.), 240
- `\bmod` (mod math op.), 169, 536
- body
 - of books, 517
 - of documents, 9, 26, 54, 176, 252–275
 - of environments, 54
 - of page, 275
 - of presentations, 323
- bold
 - font weight, 11, 104, 134, 135, 138–140, 161, 182, 199, 200, 310, 433, 546
 - math symbols, 201–202, 310
- `\boldsymbol` (math font weight com.), 182, 200, 201, 310, 543
- BOOK (bibl. entry type), 468, 473–474
- book (doc. class), xxxi, 34, 257, 417, 515–518, 520
- BOOKLET (bibl. entry type), 468, 480
- `bookmarks=true` (opt. of `hyperref` pack.), 316
- bookmarks, in PDF documents, 317
- books
 - bodies of, 517
 - document classes for, 34, 257, 515–518, 520, 522
 - options of, 518
 - final preparation of, 525–530
 - in BIB_T_EX database files, 473–474
 - logical design of, 522–525
 - numbering of structures in, 516
 - sectioning of, 516–517, 523
 - structure of, 517
 - top matter of, 255
 - writing with L^AT_EX, 515–530
- `booktabs` (pack.), 121
- `booktitle` (BIB_T_EX database field), 469, 477
- `\bot` (⊥ math sym.), 538
- bottoms, of text boxes, 449
- `\bowtie` (⋈ binary rel.), 533
- `\Box` (□ math sym.), 538
- `\boxdot` (◻ math op.), 536
- boxes, 86–96
 - and length commands, 449
 - around formulas, 209
 - commands for measuring, 450
 - invisible, *see* struts
 - multiline, 86
 - paragraph, 91–93
 - solid, 94, *see also* struts
 - text, 86–96
 - alignment of contents in, 87, 92
 - behave as single characters, 86
 - commands for, 14, 19, 69, 70, 78, 86–96, 118, 120, 152, 160–161, 179, 209, 246, 310
 - environments for, 86
 - fine tuning placement of, 95
 - framed, 90–91
 - measurements of, 449
 - single line, 86–90, 160
 - vertical alignment of, 92, 96
- `\boxminus` (⊖ math op.), 536
- `\boxplus` (⊕ math op.), 536
- `\boxtimes` (⊗ math op.), 536
- Braams, Johannes, 145, 559
- braces
 - must be balanced, 38, 58, 59, 64, 74, 178, 512
 - special, 13, 58, 59, 150, 151, 154
 - cannot overlap, 58–59
 - environments act as, 58
 - stretchable horizontal, 173–174
- braces, curly ({}), 6, 11, 14, 15, 55, 56, 58, 156, 179, 188, 189, 208, 220, 221, 287, 292, 406, 421
 - and command declarations, 407
 - as math delimiters, 43, 163, 539
 - define scope, 58–59
 - in BIB_T_EX entries, 468, 470, 472, 480
 - in `\index` commands, 512
 - in text ({}), 63, 67, 548
 - unbalanced, 38, 41, 64
 - with `\def`, 419
- brackets, square ([]), 7, 48
 - as math delimiters, 16, 163, 539

- breaking formula lines within, 221
 - enclose optional arguments, 18, 55–56, 103, 112, 115, 163, 327
 - with `\newcommand`, 410
- `\breakhere` (beamer com.), 346
- breaking
 - formulas, *see under* formulas
 - lines, *see under* lines
 - paragraphs, *see under* paragraphs
- breve (\sim text accent), 66, 546
- `\breve` (\breve math accent), 172, 543
- Bringhurst, Robert, 53, 559
- bst (BIB_TE_X style files), 465, 466, 471, 481, 483
 - determine use of fields, 469
 - form of citations created by, 472
 - location of, 483
- `\bullet` (\bullet math op.), 536
- bullet (\bullet text sym.), 67, 548
- bulleted lists, 99
- `\Bumpeq` (\asymp binary rel.), 534
- `\bumpeq` (\approx binary rel.), 534
- `\bysame` (bibl. com.), 273, 415
- `\c` (\c cedilla text accent), 66, 546
- © (copyright text sym.), 67, 548
- calc (pack.), 447, 450
- calc, TikZ library, 387
- calligraphic (math alphabet), 184, 200, 310, 433, 543
- `\Cap` (\cap math op.), 536
- `\cap` (\cap math op.), 536
- capitalization, in bibliographic databases, 273
- `\caption`, 40
 - in figures, 264
 - in tables, 27, 266
 - optional argument of, 522
- captions, 114, 522
 - in figures, 264
 - in lists, 100, 522
 - in tables, 27, 266
 - multiple, 266
- caret (^), 63
- Carlisle, David, 103, 116, 262, 265, 316, 408
- caron (~ text accent), 66, 546
- case-sensitivity
 - of BIB_TE_X fields and entry types, 469
 - of command names, 55
 - of environment names, 55
 - of `\label` arguments, 261
 - of sort keys in `\index` commands, 512
- cases (subsid. math env.), 24–25, 212, 214, 243–244
 - page breaks in, 247
- `\ccname` (redef. name), 417
- CD (subsid. math env.), 244–247
- `\cdot` (\cdot math op.), 15, 155, 536
- `\cdots` (centered ellipsis), 43, 158, 185
- cedilla (\c text accent), 66, 546
- ceiling math delimiters, 163, 539
- center (text align. env.), 7, 43, 86, 104, 114, 231, 451
 - breaking lines in, 104
 - L^AT_EX definition of, 459
- `\centerdot` (\cdot math op.), 536
- centered ellipses, 16, 157
- `\centering` (alignment com. declaration), 105, 231
- centering text, 7, 10, 43, 104
 - in columns in `tabular` environments, 115
 - in tables, 114, 117
- `centertags` (doc. class opt.), 307
- centimeter (cm abs. unit), 85, 87, 92, 447
- CERMINE, 498
- `\cfrac` (math com.), 203
- `\chapter` (struct. com.), 257, 259, 516
 - arguments of, 516
 - numbering of
 - equations in, 516
 - in front and back matter, 517
- chapter (counter), 443
- chapter (BIB_TE_X database field), 469
- `\chaptername` (redef. name), 417
 - and `hyperref` package, 319
- chapters
 - document class options to set start pages for, 518
 - grouping into parts, 516
 - in BIB_TE_X database files, 480
 - in books, 516
 - numbering of, in front and back matter, 517
- characters, *see also* glyphs
 - accented, 7, 48, 55, 65, 68, 173, 182, 317, 460, 472
 - European, 11, 65, 545–546, 549
 - following `verbatim` environments, 126
 - Greek, 162, 180, 430–432, 532
 - Hebrew, 162, 531
 - invalid, 7, 49
 - math, 12, 49
 - special, 7, 49, 63–64
 - accessing with `\symbol`, 64
 - in `\index` commands, 507
 - treating text boxes as, 86
- ChatGPT, xxv–xxvii, 419, 496, 551–558
- `\check` (\check math accent), 172, 543
- checkers, spelling, 37
- Chen, Pehong, 499, 513, 559
- `\chi` (χ Greek char.), 162, 532
- `\circ` (\circ math op.), 536
- `\circeq` (\doteq binary rel.), 534

- circle
 - draw (`\draw circle`), 362
 - draw (`\draw ellipse`), 362
- `\circlearrowleft` (\circlearrowleft math arrow), 537
- `\circlearrowright` (\circlearrowright math arrow), 537
- `\circledast` (\circledast math op.), 536
- `\circledcirc` (\circledcirc math op.), 536
- `\circleddash` (\circleddash math op.), 536
- `\circledS` (\circledS math sym.), 538
- circumflex (^)
 - text accent, 66, 546
 - text symbol, 67, 548
- citations, 42
- citations, bibliographic, 20, 259, 272, 483
 - created by BibTeX styles, 472, 482
 - hyperlinks in PDF files, 317
 - showing in margins, 262
- `\cite` (bibliographic com.), 20, 42, 259, 272
 - multiple citations, 273
 - optional argument of, 273
 - showing labels in margins, 262
 - spaces in, 273
 - with BibTeX, 482
- Citeulike, 497
- classes, document, *see* document classes
- classification of math symbols, 188–189, 199
- `\cleardoublepage` (page breaking com.), 80, 555
- `cleardoublepage.sty` (sample file), 81
- `\clearpage` (page breaking com.), 80, 81, 268
- `\cline` (table com.), 117–118
- `\clip`, TikZ command, 379
- `cls` (doc. class files), 254, 417, 454
- `\clubsuit` (\clubsuit math sym.), 538
- cm (centimeter abs. unit), 85, 87, 92, 447
- coefficients, binomial, *see* binomial coefficients
- collections, in BibTeX database files, 475–476
- `\colon` ($:$ math com.), 43, 49, 191
- colophons, of books, 517
- color
 - in presentations, 343–345
- `\color` (beamer com.), 335, 344
- `\colorbox` (beamer com.), 344
- `colorlinks` (opt. of hyperref pack.), 316, 317
- `column` (beamer env.), 343
- columns
 - double
 - document class options for, 308
 - figures spanning, 265
 - tables spanning, 266
 - in presentations, 342–343
 - math
 - adjusted, 236–247
 - adjusting with `\lalign` environment, 225
 - in aligned formulas, 223–236
 - specifying in matrix subsidiary environments, 238
 - multiple, 81
 - separator (&), 17, 115, 224, 225
 - width of, in tabular environments, 116
- `columns` (beamer env.), 343
- combining
 - encapsulators in `\index` commands, 504
- command declarations, 60, 104, 105, 117, 231
 - affecting multiple paragraphs, 136
 - do not take arguments, 60
 - for fonts, *see* font command declarations
 - scope of, 135, 407–408, 423
 - using in custom commands, 136, 407–408
- command files, *see also* sty files and packages, 254, 406, 427–436
- commands in, 427
- terminating, 436
- commands, 54–61, 193
 - *-ed form, 55, 77, 80, 83, 85, 109, 110, 128, 198, 200, 247, 257, 318, 413, 416, 423, 426, 483, 517, 519
 - arguments of, *see under* arguments or specific commands, 323
 - custom, *see* custom commands
 - defining, *see* custom commands
 - delimited, *see* delimited commands
 - for boxes, *see under* boxes
 - for expanding values, 66–68, 72, 74, 94, 445
 - for filling horizontal space, 85–86, 94, 106, 184, 238, 240, 458
 - for fonts, *see* font commands
 - for graphics, *see under* graphics
 - for indexing, *see under* indexing
 - for presentations, *see under* presentations, beamer, *see under* presentations, beamer, and FoilTeX
 - for tables, *see under* tables or floats
 - fragile, 60–61
 - protecting, 60, 256, 257, 516, 521
 - generalized, *see* custom commands
 - global, *see under* scope
 - in preamble, *see under* preamble
 - length, *see* length commands
 - local, *see under* scope
 - logical design using, 37
 - long, 60
 - names of, 54
 - case-sensitivity of, 55
 - primitive, *see* primitive commands
 - providing, 415
 - redefining, 414–415
 - scope of, *see under* scope
 - sectioning, *see* structural commands

- short, 60, 135, 281, 292
 - defining, 413–414
 - spacing, *see* spacing commands
 - start with `\`, 6, 11, 54
 - structural, *see* structural commands
 - terminating, 14, 55, 152
 - \TeX , *see under* \TeX
 - text style, *see* text style commands
 - to avoid using, 140, 525
 - types of, 60–61
 - user-defined, *see* custom commands
- commas (,), 7, 48
 - and italic correction, 137
 - in `\bibitem` commands, 273
 - in \LaTeX databases, 468, 469, 488
 - use in formulas, 153
- `comment` (comment env.), 73–74, 428
 - locating errors with, 74
 - nesting, 73
- comments, 8, 71–74
 - block comment, 73
 - creating with % characters, 72, 73, 121, 303
 - creating with `comment` environments, 73–74
 - finding errors using, 39, 72, 406, 428
 - in \LaTeX database files, 72, 487
 - inserted using `kill` command, 121
 - marginal, 93–94, 527
 - in double-sided documents, 93
 - in equations, 93
 - in multiline math environments, 93
 - space between, 448
 - width of paragraph box for, 94
 - removing end-of-line characters with, 71, 72, 508
- common errors, 41–44
- commutative diagrams, 244–247, 310
- `\complement` (\mathbb{C} math sym.), 538
- components
 - building formulas from, 176–178
 - of formulas, 14–19, 154–178
 - of pages, 275
- Computer Modern (CM) fonts, 134, 138, 143
 - bold extended, 134
 - ligatures in, 64
 - math bold italic, 200
 - math italic, 199
 - PostScript versions of, 33, 143
 - roman, 134, 199
 - sans serif, 134
 - typewriter, 64, 65, 135
- conference proceedings, in \LaTeX database files, 468, 474–476
- `\cong` (\cong binary rel.), 533
- congruences, 16, 169
 - as binary operations, 169
- consecutive numbering, of proclamations, 107–108
- contents, table of, *see under* tables
- `\contentsline` (toc file com.), 520
- `\contentsname` (redef. name), 417
- context, affects style of emphasized text, 136
- continued fractions, 203
- `\contrib` (top matter com.), 284
- controls, float, 114, 267–268
- converting
 - from articles to presentations, 322
- `\coordinate`, `TikZ` command, 378
- `\coprod` (\prod large math op.), 171, 542
- `\copyright` (\copyright copyright), 67, 548
- corner math delimiters (\llcorner , \lrcorner , \ulcorner , \urcorner), 163, 539
- corollaries, *see* proclamations
- correction, italic, *see* italic correction
- `\cos` (cos math op.), 168, 540
- `\cosh` (cosh math op.), 168, 540
- `\cot` (cot math op.), 168, 540
- `\coth` (coth math op.), 168, 540
- counters, 103, 238, 419, 443–447
 - and `\label` commands, 446
 - and proclamations, 443
 - arithmetic with, 446–447
 - defined in preamble, 444
 - defining, 444
 - incrementing, 446
 - \LaTeX , 443
 - linking with other counters, 444, 445
 - names of, 108
 - store numbers, 443
 - styles for, 74, 103, 424, 445–446, 455
 - values of, 36, 74, 445
 - printing, 66–68, 94, 445
 - setting, 443–444, 446
 - with `list` environments, 455
- Courier, in \LaTeX documents, 144
- `covers.pdf` (sample file), 27
- cross-referencing, 259–264, 274, 318
 - and automatic renumbering, 20, 36
 - in indexes, 507
 - in proof environment titles, 112
 - inside presentations, 324
 - labels for, 20, 23, 27, 39, 42, 72, 207, 219, 235, 256, 259, 263, 266, 274, 446
 - of equations, 19–21, 23, 42, 153, 207, 223, 259
 - groups of, 222–223
 - in multiline math environments, 219
 - of list items, 99, 103
 - of tables, 114
 - page numbers, 20, 23, 264, 274, 527

- section numbers, 23, 103, 112, 114, 153, 219, 256, 259, 310
 - using ties (\sim) with, 52, 262
 - within BibTeX databases, 476–477
 - `\crossref` (BibTeX database field), 469
 - `CrossTeX`, 497
 - Crémer, Jacques, 559
 - `\csc` (csc math op.), 168, 540
 - CTAN (Comprehensive T_EX Archive Network)
 - packages on, 514
 - CTAN (Comprehensive T_EX Archive Network), 143
 - `\Cup` (\mathbb{U} math op.), 536
 - `\cup` (\cup math op.), 536
 - `\curlyeqprec` (\curlyeqprec binary rel.), 534
 - `\curlyeqsucc` (\curlyeqsucc binary rel.), 534
 - `\curlyvee` (\vee math op.), 536
 - `\curlywedge` (\wedge math op.), 536
 - `\curraddr` (top matter com.), 285
 - rules for using, 285
 - `\curvearrowleft` (\curvearrowleft math arrow), 192, 537
 - `\curvearrowright` (\curvearrowright math arrow), 192, 537
 - curved line
 - draw (`\draw dot`), 365
 - custom commands, 41, 404–421, 523
 - arguments of, 410–414
 - as shorthand, 404–414
 - command declarations in, 136
 - scope of, 407–408, 423
 - defining, 74, 78, 112, 119, 273, 274, 404, 406, 413–417, 445
 - in preamble, 254
 - delimited, 419–421
 - for environments, 406
 - for indexing, 411, 504
 - for subformulas, 411–413
 - for text, 406
 - invoking, 406
 - multiple arguments of, 56
 - optional arguments of, 414
 - rules, 406, 407
 - short arguments, 413–414
 - with `\def`, 419
 - custom environments, 421–427, 523, 525
 - defining, 422
 - for custom lists, 451–459
 - optional arguments of, 425–426
 - short arguments, 426
 - customizing, 404, 427
 - article templates
 - for AMS document classes, 303–306
 - preambles of, 303
 - top matter of, 303
 - indentation of text, 83
 - L^AT_EX, 403–461
 - dangers of, 459–461
 - lists
 - and counters, 455
 - defining new, 451–459
 - theorem styles, 111
- `\d` ($\underset{\cdot}{\text{d}}$ underdot text accent), 66, 546
- `\dag` (\dagger dagger)
 - math symbol, 538
 - text symbol, 67, 548
- `\dagger` (\dagger math op.), 536
- `\daleth` (\daleth Hebrew char.), 162, 531
- dashes, 48
 - em dash (---), 11, 62, 67, 548
 - en dash (--), 11, 62, 67, 548
 - in BibTeX databases, 471
- `\dashleftarrow` (\dashleftarrow math arrow), 537
- `\dashrightarrow` (\dashrightarrow math arrow), 537
- `\dashv` (\dashv binary rel.), 533
- databases, BibTeX, *see also* bib, 465, 467–481, 483
- datavisualization, TikZ library, 394
- `\date` (top matter com.), 66, 253, 283, 312, 518
- `\datename` (redef. name), 417
- dates
 - commands for, 10, 57, 66–68, 145
 - L^AT_EX release, 72, 254
- Davey, Brian, 41
- `\day` (time com.), 66
- `\dbinom` (display-style binomial com.), 157
- `\dcolumn` (pack.), 116
- `\ddag` (\ddagger dagger)
 - math symbol, 538
 - text symbol, 67, 548
- `\ddagger` (\ddagger math op.), 536
- `\ddddot` ($\ddddot{\text{x}}$ math accent), 172, 543
- `\dddots` (\dddots math accent), 172, 543
- `\ddot` ($\ddot{\text{x}}$ math accent), 172, 543
- `\ddots` (\ddots diagonal ellipsis), 159, 240
- decimals, alignment on, 116
- declarations, command, *see* command declarations
- `\DeclareMathOperator` (op. def. com.), 42, 198, 200, 202, 254, 310, 415
- `\DeclareMathOperator*` (op. with limits def. com.), 198, 416
- declaring, types of math symbols, 198
- `\dedicatory` (top matter com.), 282
- `\def` (T_EX com.), 460
 - defining commands with, 41, 419
 - overwriting previously defined commands with, 419
- definition (procl. style), 109–111
- `\deg` (deg math op.), 168, 540

- delimited commands, 419–421
 - invoking, 420
- delimiters
 - for displayed math environments, 14, 19, 43
 - for inline math environments, 13, 60, 150–151
 - for `\verb` commands, 127–128
 - in L^AT_EX database files, 468
 - math, *see* math delimiters
- `\Delta` (Δ Greek char.), 162, 532
- `\delta` (δ Greek char.), 162, 532
- depth
 - of tables of contents, 447
 - of text boxes, 88, 92, 449
- `\depth` (length com.), 87, 88, 92
- `description` (text list env.), 98, 100–101
- design
 - logical, 37, 251
 - of books, 522–525
 - visual, 37, 251, 275–278, 353
- `\det` (det math op.), 168, 540
- `\dfrac` ($\frac{x}{y}$), 155, 184, 203
- `\DH` (Eth Eur. char.), 549
- `\dh` (eth Eur. char.), 549
- `\diagdown` (\diagdown math sym.), 538
- diagonal ellipses, 240
- diagrams, commutative, *see* commutative diagrams
- `\diagup` (\diagup math sym.), 538
- `\Diamond` (\Diamond math sym.), 538
- `\diamond` (\diamond math op.), 536
- `\diamondsuit` (\diamondsuit math sym.), 538
- dieresis, *see* umlaut
- diesis (§)
 - math symbol, 536, 538
 - text symbol, 67, 548
- `\digamma` (\digamma Greek char.), 162, 532
- `\dim` (dim math op.), 168, 540
- dimensions, *see also* measurements
 - units for measuring, 12, 76, 85, 87, 92, 134, 138, 142, 190, 447, 544
 - with length commands, 448
- directories, *see under* AMS and L^AT_EX distributions
- `\displaybreak` (display break com.), 247, 248
 - optional argument of, 247
- displayed formulas, *see* displayed math environments
- displayed math environments, *see* displayed math environments
- displayed fractions, *see under* fractions
- displayed math environments, 17, 19–25, 43, 150
 - `\[` (start math mode), 14, 43, 150
 - Visual Guide* to, 211, 212
 - `\]` (end math mode), 14, 43, 150
 - act as special braces, 150
 - align, 21–24, 43, 212, 214, 223–228, 234, 235, 247
 - `align*`, 23, 225, 231
 - `alignat`, 43, 212, 214, 228–230
 - blank lines in, 39
 - breaking pages in, 247–248
 - `displaymath`, 150, 151, 153, 154, 235
 - `eqnarray`, 226–227
 - equation, 19, 24, 153–154, 205, 235, 324
 - `equation*`, 19, 154, 205, 235, 324
 - `flalign`, 212, 214, 225, 235
 - font size in, 203
 - `gather`, 43, 212, 214–216, 223, 235, 247
 - `gather*`, 215
 - in direct succession, 526
 - inline-style binomials in, 157
 - `multline`, 212, 214, 216–218
 - `multline*`, 217–218
 - subequations, 206, 207, 223, 445
- displayed text environments, 97, 123–127
 - `alltt`, 157
 - blank lines in, 98
 - `multicols`, 81
 - quotation, 124
 - `quote`, 43, 123
 - tabbing, 121–123
 - `tabular`, 64, 113–120, 242, 266
 - `theorem`, 37, 105, 108, 338, 425
 - `verbatim`, 125–127, 129
 - `verse`, 124–125
 - `xcb`, 522
- `displaymath` (disp. math env.), 150, 151, 153, 154, 235
- `\displaystyle` (math font size com.), 203, 207
- dissertations, in L^AT_EX database files, 477–478
- distributions
 - AMS packages, 279, 309–311, 321, 421, 461, 480, 481
 - L^AT_EX, 72, 262, 312
 - PSNFSS, 143
- `\div` (\div math op.), 155, 536
- `\divideontimes` (\oslash math op.), 536
- division, 15, 49, 155, 191, 536
- `\DJ` (Dyet Eur. char.), 549
- `\dj` (dyet Eur. char.), 549
- `document` (text env.), 9, 54, 176, 252
- document class options
 - 9pt, 306
 - 10pt, 84, 306
 - 11pt, 306
 - 12pt, 138, 306
 - `a4paper`, 307
 - are passed on to packages, 254
 - `centertags`, 307
 - `draft`, 76, 253, 309

- final, 309
- fleqn, 151, 307
- for AMSFonts, 309
- for double-column documents, 308
- for double-sided printing, 308, 518
- for font sizes, 306
- for position of equation numbers, 307
- for start of chapters, 518
- for title pages, 308, 518
- handout, 353
- legalpaper, 71, 307
- leqno, 307
- letterpaper, 307
- noamsfonts, 309
- nomath, 309
- notitlepage, 308, 518
- onecolumn, 308
- oneside, 308, 518
- openany, 518
- openright, 518
- reqno, 307
- tbtags, 307
- titlepage, 253, 308, 518
- trans, 353
- twocolumn, 71, 81, 265, 266, 308
- twoside, 71, 253, 308, 518
- document classes, 37, 68, 236, 417
 - amsart, 25–311, 515, 518
 - amsbook, xxxi, 34, 257, 515, 518, 522
 - amsproc, 281
 - anatomy of, 25
 - article, 26, 34, 193, 253, 255, 258, 311, 312
 - beamer, 321–355
 - book, xxxi, 34, 257, 515, 518, 520
 - cls files, 254, 417, 454
 - determine
 - placement of equation numbers, 307
 - position of equations, 307
 - spacing, 81
 - for books, 34, 257, 515–518, 520, 522
 - options of, 518
 - for presentations, *see under* beamer
 - indexing with AMS, 500
 - legacy, 98, 253, 258, 311, 312
 - letter, 312
 - report, 257, 312
 - sample, 8
 - sample.cls, 9
 - slides, 312
- document font families, 6, 11, 104, 134–135, 138, 144, 161, 310, 433, 456, 546
 - normal, 135
 - roman, 135, 143
 - sans serif, 135, 143
 - typewriter style, 135, 143
- documentation
 - for beamer document class, 322, 354
- \documentclass (preamble com.), 5, 25, 56, 71, 179, 244, 253, 303, 322, 350
 - and \NeedsTeXFormat command, 254
- documents
 - body of, 9, 26, 54, 176, 252–275
 - design of, 275–278
 - L^AT_EX, 251–278
 - on the Internet, 319
 - using AMS document classes in, 279
 - preamble of, 253–254
 - printing and viewing, 6
 - readability of, 138, 353
- dollar sign (\$), *see also* \$ and \\$, 7
 - as inline math delimiter, 12, 13, 60, 150, 548
 - as text symbol, 63, 67
- Doob, Michael, 91, 559
- dot
 - draw (\draw dot, 364
- \dot (*i* math accent), 172, 543
- \doteq (\doteq binary rel.), 533, 534
- \dotfill (space fill com.), 85
- dotless i and j (*i* and *j*), 65, 66, 546
- \dotplus ($\dot{+}$ math op.), 536
- dots
 - filling lines with, 85
- \dots (... math ellipsis com.), 16, 43, 64, 157–159, 185
- \dotsc (... math com.), 158
- \dotsc (... math com.), 158
- \dotsi (... math com.), 158
- \dotsm (... math com.), 158
- \dotso (... math com.), 158
- double accents, in math, 173
- double acute (ˆ text accent), 66, 546
- double dagger (‡)
 - math symbol, 536, 538
 - text symbol, 67, 548
- double guillemet, 549
- double quote ("), *see also* quotation marks, 7, 61, 67, 548
 - in \index commands, 507
 - in B^IB^TE_X database fields, 489
 - key, 7, 49, 61
- double spacing, 78
- double subscripts and superscripts, font size of, 203
- double vertical lines in CD environments, 246
- double-column
 - documents
 - document class options for, 71, 308
 - figures and tables, 265, 266

- double-sided printing
 - and marginal comments, 93
 - document class options for, 71, 308, 518
- `\doublebarwedge` ($\overline{\wedge}$ math op.), 536
- `\doublecap` (\cap math op.), 536
- `\doublecup` (\cup math op.), 536
- `\doublespacing` (`setspace` com.), 78
- `\Downarrow` (\Downarrow)
 - math arrow, 537, 539
 - math delimiter, 163
- `\downarrow` (\downarrow)
 - math arrow, 537
 - math delimiter, 163, 539
- `\downdownarrows` (\Downarrow math arrow), 537
- Downes, Michael, 491
- `\downharpoonleft` (\lrcorner math arrow), 537
- `\downharpoonright` (\llcorner math arrow), 537
- `draft` (doc. class opt.), 76, 253, 309
- draw
 - Bézier cubics (`\draw Bézier cubics`), 367
 - circle (`\draw circle`), 362
 - curved line (`\draw dot`), 365
 - dot (`\draw dot`), 364
 - ellipse (`\draw ellipse`), 362
 - grid (`\draw grid`), 370
 - rectangle (`\draw rectangle`), 369
- `\draw`, TikZ command, 358–397
 - parameters, 360–362
- drivers
 - printer, 36
 - video, 36
- Duma, Jacques, 561
- Dupré, Lyn, 71, 559
- `dvipsnam` (opt. of `xcolor` pack.), 344
- dyet (Eur. char), 549
- editing cycle, 5, 6
- edition (BIBTeX database field), 469
- editor (BIBTeX database field), 469
- editors, text, 5, 6, 13, 34, 38, 49, 70, 133, 405
- electronic mail address, *see* `\email`
- `\ell` (ℓ math sym.), 538
- ellipses (\dots)
 - centered (\cdots), 16, 158
 - diagonal, 240
 - in math, 16, 157–159, 238
 - in text, 64
 - vertical, 240
- `\em` (font shape com. dec.), 104, 135, 136, 138, 408, 423, 546
- `em` (font shape env.), 104
- `em` (rel. unit), 447, 544
- `em dash` (---), 11, 62, 67, 69, 548
- `\email` (top matter com.), 285–286, 289–291, 304, 305
 - rules for using, 285–286
- `\emph` (font shape com.), 6, 11, 54, 135, 136, 334, 546
- emphasized (font shape), 6, 11, 20, 54, 104, 106, 135–137, 153, 154, 408, 423, 546
 - and italic correction, 138
 - context dependence of, 136
- empty (page style), 275
- empty group (`{ }`), 57, 156, 175, 188, 189, 197, 208, 287
- `\emptyset` (\emptyset math sym.), 538
- `en dash` (--), 11, 67, 69, 548
 - in BIBTeX databases, 471
- `\enclname` (redef. name), 417
- encodings, font, *see* font encodings
- `\end` (end of env.), 54, 58
 - errors with, 58, 127
- `\endinput` (input ending com.), 436
- eng (Eur. char), 549
- `\enlargethispage` (spacing com.), 80, 527, 530
- `\enlargethispage*` (spacing com.), 80
- `\ensuremath` (math mode com.), 409, 410, 415
- Enter key, *see* Return key
- entries
 - glossary, *see under* glossaries
 - index, *see under* indexing
- `enumerate` (list text env.), 424
- `enumerate` (pack.), 103, 424, 428
- `enumi` (counter), 103, 443
- `enumii` (counter), 443
- `enumiii` (counter), 443
- `enumiv` (counter), 443
- environments, 6, 10, 13, 54–61
 - *-ed forms, 154, 205, 215, 217, 219, 235, 265
 - act as braces, 58
 - arguments of, *see under* arguments or specific environments
 - `\in`, 104, 125
 - begin with `\begin`, 54, 58
 - body of, 54
 - case-sensitivity of names of, 55
 - custom, *see* custom environments, *see* custom environments
 - end with `\end`, 54, 58
 - font size, 104
 - for presentations, *see under* beamer and FoilTeX
 - for tables, *see* tables
 - indenting contents of, in source file, 153
 - legacy, 123–125, 451
 - list, *see* list text environments, *see* list text environments
 - math, *see under* inline and displayed math

- environments, subsidiary math environments, *and* displayed text environments modifying, 422–424
- `\newline` in, 78
- short, 426
- subsidiary math, *see* subsidiary math environments
- `\epsilon` (ϵ Greek char.), 162, 532
- `\eqcirc` ($=$ binary rel.), 534
- `\eqarray` (math align. env.), 226–227
- `\eqref` (cross-ref. com.), 20, 23, 42, 153, 154, 207, 219, 223, 259, 260, 272, 410
- `\eqslantgtr` (\gtrless binary rel.), 534
- `\eqslantless` (\lessgtr binary rel.), 534
- equality ($=$)
 - binary relation, 49, 533
 - in L^AT_EX database fields, 468
 - key, 7
- equation (counter), 443
- equation (disp. math env.), 19, 24, 153–154, 205, 235, 324
 - blank lines in, 154
- `equation*` (disp. math env.), 19, 154, 205, 235, 324
- `\equationname` (`hyperref` redef. name), 319
- equations, 19–21
 - document class options for placement of numbers, 307
 - grouping, 206–207
 - in multiline math environments, 219
 - labels for, 20, 259
 - marginal comments in, 93
 - numbering of, 19–21, 42, 153–154, 205, 215, 217, 219
 - groups, 222–223
 - in chapters of books, 516
 - preventing, 22, 23, 154, 205, 215, 217, 219, 222, 235
 - variants, 206, 222
 - within sections, 19, 154, 254
 - position of, 307
 - systems of, 229–230
 - tagging, 21, 205–207, 215, 217, 219, 222, 234, 235
- `\equiv` (\equiv)
 - binary relation, 533
 - math operator, 16, 169, 180, 412, 421
- error messages, 37–40
 - math, 40, 41
 - Missing `\right.` inserted, 221
- errors
 - common, 41–44
- eszett (β , SS), 67, 145, 545
- `\eta` (η Greek char.), 162, 532
- `\eth` (\eth math sym.), 538
- eth (Eur. char), 549
- `\eucal` (pack.), 310, 311, 427, 543
 - options, 310, 433, 543
- `\eufrak` (pack.), 310
- Euler Fraktur (math alphabet), 179, 201, 310, 433, 543
- Euler Script (math alphabet), 310, 311, 433, 543
- European
 - accents, 11, 65
 - characters, 11, 65, 545–546, 549
 - quotation marks, 549
- ex (rel. unit), 447
- examples
 - of bibliographies, 268–271, 465–467
 - of indexes, 500–502
 - of nonbreakable spaces (`\~` spacing com.), 62
 - of top matter commands, 289–292
- `\except` (table of contents com.), 519
- exclamation marks (!), 7, 48
 - as float control, 267, 529
 - in `\index` commands, 503, 504, 507
 - Spanish (¡), 67, 545, 548
 - terminating sentences with, 50–53
- exercises, 522
 - in books, 522
 - within a section, 522
- `\exists` (\exists math sym.), 538
- `\exp` (exp math op.), 168, 540
- expanding values of counters, 66, 68, 445
- `\exscale` (pack.), 310
- extensibility of L^AT_EX and T_EX, 33
- `\fallingdotseq` (\fallingdotseq binary rel.), 534
- `\fancyhdr` (pack.), 278
- `\fboxrule` (length com.), 91
- `\fboxsep` (length com.), 91
- `\fcolorbox` (beamer com.), 344
- `\fd` (font def. file), 145
- Fear, Simon, 121
- fields, bibliographic, *see under* bib
- figure (counter), 443
- figure (float env.), 264–266
 - optional arguments of, 267, 529
 - placement of, 527, 529
- `figure*` (float env.), 265
- `\figurename` (redef. name), 417
 - and `hyperref` package, 319
- figures, 264–266
 - captions in, 522
 - fragile commands in, 60
 - commands for, 264, 265, 522
 - double-column, 265
 - forcing typesetting of, 81
 - lists of, *see also under* lists, 521–522

- adding a line to, 522
 - fragile commands in, 60
 - numbering of, 264, 265
 - old-style, 48
 - placement of, 267, 527, 529
- file formats, PDF, for graphics, 265
- files
 - MakeIndex* log, *see* ilg
 - auxiliary, *see* aux
 - BIB \TeX log, *see* blg
 - BIB \TeX style, *see* bst
 - class (cls), *see under* document classes
 - command, *see* command files
 - converting from articles to presentations, 322
 - glossary, *see* glo
 - hyperref bookmarks, *see* out
 - including in other files, 427
 - index entry, *see* idx
 - list
 - of figures, *see* lof
 - of tables, *see* lot
 - listing those used by a document, 428
 - log, *see* log files
 - package source, *see under* packages
 - Portable Document Format, *see* PDF
 - processed index, *see* ind
 - sample, *see* sample files
 - source, *see* source files
 - style, *see* sty
 - tables of contents, *see* toc
 - terminating, 436
- \fill, TikZ command, 365
- \filldraw, TikZ command, 365
- filling horizontal space, 85–86, 94, 106, 184, 238, 240, 458
- final (doc. class opt.), 309
- final preparation, of books, 525–530
- fine tuning
 - of mathematical formulas, 190–192
 - placement of root with \sqrt, 160
 - placement of text boxes, 96
- \Finv (∇ math sym.), 538
- firstpresentation.tex (sample file), 28
- Fischer, Ulrike, 560
- fixed-size math delimiters, 165
- flalign (math align. env.), 212, 214, 225, 235
- \flat (\flat math sym.), 538
- fleqn (doc. class opt.), 151, 307
- floats, 264–268
 - commands, 27, 266, 267
 - controls, 114, 267–268
 - environments, 113, 264–266
 - figure, 264–267
 - figure*, 265
 - locating, 527, 529
 - table, 114, 264, 267
 - table*, 266
 - forcing typesetting of, 81
 - specifying placement of, 267, 527
- floor math delimiters, 163, 539
- flush left
 - alignment of formulas, 211, 214, 216, 218, 225
 - column alignment, 236–237, 241
 - columns in tabular environments, 115
 - setting equations, 307
- flush right
 - alignment of formulas, 211, 214, 216, 218, 225
 - alignment of text, 86
 - column alignment, 236–237, 241
 - columns in tabular environments, 115
 - setting equations, 307
- flushleft (text align. env.), 104, 451
 - breaking lines in, 104
- flushright (text align. env.), 54, 104, 451
 - breaking lines in, 104
- \fnsymbol (footnote counter style com.), 74
- font command declarations, *see also* font commands, 136
 - for shape
 - \em, 104, 135, 136, 138, 546
 - \itshape, 104, 135–139, 456, 546
 - \normalfont, 103, 135, 144, 161, 546
 - \rmfamily, 104, 135, 546
 - \scshape, 104, 135, 136, 546
 - \sffamily, 104, 135, 139, 546
 - \slshape, 104, 135, 136, 138–140, 546
 - \ttfamily, 104, 123, 135, 546
 - \upshape, 104, 135, 136, 424, 456, 546
 - for weight
 - \bfseries, 58–60, 104, 135, 138–140, 546
 - \mdseries, 135, 138, 546
- font commands, *see also* font command declarations
 - and italic correction, 135, 138
 - and *MakeIndex*, 136
 - for selecting fonts using family names, 144
 - for series
 - \textmd, 135, 138, 546
 - for shape
 - \emph, 11, 54, 135, 136, 334, 546
 - \textit, 135, 136, 334, 546
 - \textnormal, 135, 161, 546
 - \textrm, 135, 310, 546
 - \textsc, 135, 136, 546
 - \textsf, 135, 546
 - \textsl, 135, 136, 546
 - \texttt, 11, 63, 135, 546

- `\textup`, 135, 136, 546
- for size
 - `\footnotesize`, 138, 139, 547
 - `\Huge`, 138, 139, 547
 - `\huge`, 138, 139, 547
 - `\LARGE`, 138, 139, 547
 - `\Large`, 138, 139, 547
 - `\large`, 138, 139, 547
 - `\larger`, 139
 - `\normalsize`, 138, 139, 547
 - `\scriptsize`, 138, 139, 547
 - `\SMALL`, 138, 139, 547
 - `\Small`, 138, 139, 547
 - `\small`, 138, 139, 547
 - `\Tiny`, 138, 139, 547
 - `\tiny`, 138, 139, 547
- for weight
 - `\textbf`, 11, 60, 135, 138, 161, 310, 334, 546
- low-level, 142
- math, *see* math font commands
- obsolete, 140
- orthogonality of, 139–140
- two-letter (obs. L^AT_EX 2.09), and italic correction, 140
- using in math environments, 161
- font encodings, 64, 65, 141
 - koi8-u, 146
 - LY1, 145
 - T1, 69, 549
- font environments
 - for series
 - `bfseries`, 104
 - for shape
 - `em`, 104
 - `itshape`, 104
 - `rmfamily`, 104
 - `scshape`, 104
 - `sffamily`, 104
 - `slshape`, 104
 - `ttfamily`, 104
 - `upshape`, 104
 - for weight
 - `bfseries`, 104
- font families, document, *see* document font families
- font metrics files, 144
- font series, 134, 138
 - command declarations for, *see* command declarations
 - commands for, *see under* font commands
 - environments for, *see under* font environments
- font shapes
 - command declarations for, *see under* font command declarations
 - command declarations
 - commands for, *see under* font commands
 - environments for, *see under* font environments
 - sans serif, 134
 - serif, 134
- font substitution, 139, 527
 - warning messages, 139
- font weight, 134
 - command declarations for, *see under* font command declarations
 - command declarations
 - commands for, *see under* font commands
 - environments for, *see under* font environments
- font width, 134
- `fontenc` (pack.), 145, 146
- `\fontfamily` (font selection com.), 144
- fonts
 - AMSFonTS, 309–311
 - document class options for, 309
 - blackboard bold, 180, 201, 433
 - bold math, 182, 199, 201, 543
 - calligraphic, 184, 200, 310, 433, 543
 - commands, *see under* font commands *and* font command declarations
 - Computer Modern, *see* Computer Modern
 - Courier (typewriter shape), 143, 144
 - encodings, *see* font encodings
 - Euler Fraktur, 179, 201, 310, 433, 543
 - Euler Script, 310, 311, 433, 543
 - for text in math mode, 161
 - Helvetica, 135, 143, 144
 - Lucida Bright, 143–145
 - math, 199–201
 - size of, 203
 - monospaced, 133, 134
 - proportional, 133, 134
 - selecting, 133–146
 - with `\fontfamily` command, 144
 - with `\selectfont` command, 142, 144
 - shapes, *see* font shapes
 - size of, 42, 104, 134, 138–547
 - document class options for, 306
 - sources for, 143
 - STIX, 193
 - substitution of, 139, 527
 - Times, 135, 143
 - typewriter style, 133, 134
 - weight, *see* font weight
 - width, *see* font width
- `fonttbl.tex` (sample file), 64, 114
- footers, page, 275–277
- `\footnote`, 74
- `footnote` (counter), 443

- footnotes, 74–75
 - fragile commands in, 60
 - indicated by symbols, 74
 - on title pages, 74
 - unmarked, 287
 - `\footnotesize`, 138, 139, 547
 - `\forall` (math sym.), 538
 - forcing
 - floats to typeset, 81
 - indentation, 79
 - `\foreach`, TikZ command, 374
 - formatting of documents, determined by document classes, 47
 - `formulanote.tex` (sample file), 12–14, 40–41
 - formulas, 6, 13
 - adjusted, 214, 236–247
 - flush left and right, 218
 - aligning, 21–24, 211–214, 223–236, 247, 307
 - multiline, 221–222
 - rules for, 221–222, 225
 - text in, 230–231
 - annotating, 229
 - boxed, *see under* boxes
 - breaking into multiple lines, 22–23, 221–222
 - displayed, 43
 - rules for, 221
 - commas in, 153
 - components of, 14–19, 154–160
 - building using, 176–178
 - displayed, *see* displayed math environments
 - see* displayed math environments
 - gallery, 178–185
 - grouping, 214–216
 - inline, *see* inline math environments
 - multiline, 21–24, 43
 - splitting, 216–218
 - numbering, *see under* equation
 - specifying fonts for, 161
- foundries, type, 143
- Fowler, H. W., 530, 559
- `\frac` ($\frac{x}{y}$), 14, 15, 56, 155, 184, 203
- fractions, 14, 15, 56, 155, 184
 - continued, 203
 - displayed, 155, 184
 - generalized, 207–209
 - inline, 155
- fragile commands, 60–61
 - protecting, 60, 256, 257, 516, 521
- `\frame` (beamer env.), 29–31, 323, 324, 339
- `\framebox`, 90
- frames, *see also under* presentations, 323, 353
 - environments of, 323
 - outline (table of contents), 324
 - titles, 29–31, 353
- `\frametitle` (beamer com.), 29–31, 323, 324, 339
- `\frenchspacing` (spacing com.), 53
- front matter, 517
 - numbering of chapters in, 517
 - of AMS documents, 292
 - of books, 519–521
- `\frontmatter` (struct. com.), 517
- `\frown` (\curvearrowright binary rel.), 533
- `\Game` (\oslash math sym.), 538
- `\Gamma` (Greek char.), 162, 532
- `\gamma` (γ Greek char.), 162, 532
- `gather` (disp. math env.), 43, 212, 214–216, 223, 235, 526
 - page breaks in, 247
 - rules for, 215
- `gather*` (disp. math env.), 215
- `gathered` (subsid. math env.), 214, 232–235
 - and `\allowdisplaybreaks` commands, 248
- `\gcd` (gcd math op.), 168, 540
- `\ge` (\geq binary rel.), 533
- generalized
 - commands, *see* custom commands
 - fractions, 207–209
- `\genfrac` (math com.), 207–209
- geometry (pack.), 43, 277
- `\geq` (\geq binary rel.), 533
- `\geqq` (\geq binary rel.), 534
- `\geqslant` (\geq binary rel.), 534
- german (opt. of babel pack.), 145
- `german.tex` (sample file), 145
- `\gg` (\gg binary rel.), 533
- `ggamsart.tpl` (template file), 306
- `\ggg` (\ggg binary rel.), 534
- `\gimel` (\gimel Hebrew char.), 162, 531
- `glo` (glossary files), 514
- global commands, *see under* scope
- glossaries, 514
 - as a custom list environment, 458
 - auxiliary file (`glo`), 514
 - defining, 514
- `glossaries` (pack.), 514
- `\glossary` (glossary com.), 514
- glue, 419, 451
 - vertical, *see also under* vertical spacing
- glyphs, *see also* characters
- `\gnapprox` (\gtrapprox neg. binary rel.), 535
- `\gneq` (\gtr neg. binary rel.), 535
- `\gneqq` (\gtr neg. binary rel.), 535
- `\gnsim` (\gtrsim neg. binary rel.), 535
- Google, 514
- Google Scholar, 497

- Goossens, Michel, 559
- GPT SP, 554, 555, 557
- GPT SP category
 - Programming, 555
 - Writing, 554
- GPT Store, 554
- graphics, 27
 - commands for, 265, 266, 337
 - in presentations, 338
 - formats
 - PDF, 265
 - including in documents, 264–266
 - Inkspace, 28
 - scaling, 265
- graphicx (pack.), 27, 192, 254, 265–266, 309
- grave ($\grave{\text{~}}$ grave text accent), 48, 66, 546
- \grave ($\grave{\text{~}}$ math accent), 172, 543
- greater than ($>$)
 - as binary relation, 533
 - text symbol, 67, 548
- Greek letters, 162, 180, 430–432, 532
- Greenwade, George D., 78
- grid
 - draw (\draw grid, 370
- grouping
 - chapters into parts, 516
 - equations, 206–207
 - formulas, 214–216
- groups
 - of equations, 206–207
 - cross-referencing, 222–223
 - labels for, 207
 - numbering, 222–223
- \gtrapprox (\gtrapprox binary rel.), 534
- \gtrdot (\gtrdot binary rel.), 534
- \gtreqless (\gtreqless binary rel.), 534
- \gtreqqless (\gtreqqless binary rel.), 534
- \gtrless (\gtrless binary rel.), 534
- \gtrsim (\gtrsim binary rel.), 534
- guillemets, 549
- \guillemotleft (Eur. quot. mark), 549
- \guillemotright (Eur. quot. mark), 549
- \guilsingleft (Eur. quot. mark), 549
- \guilsingright (Eur. quot. mark), 549
- Gurari, Eitan, 559
- \gvertneqq (\gvertneqq neg. binary rel.), 535
- \H double acute text accent (¨), 66, 546
- Hahn, Harley, 560
- handout (beamer doc. class opt.), 353
- \hangafter
 - \TeX command, 418
 - length command, 79
- \hangindent (length com.), 79
- hanging indentation, 79–80, 125
- Harrison, Michael A., 499, 513, 559
- Hart, Horace, 560
- \hat ($\hat{\text{~}}$ math accent), 17, 172, 543
- \hbar (\hbar math sym.), 538
- \hdotsfor (space fill com.), 184, 238, 240
 - optional argument of, 238
- headers, page, *see* running heads
- headings (page style), 277
- \headtoname (redef. name), 417
- \heartsuit (\heartsuit math sym.), 538
- Hebrew letters, 162, 531
- \height (length com.), 87, 88, 92
- height, of text boxes, 88, 91, 92, 94, 449
- Helvetica (sans serif font), 135, 143
- \hfill (space fill com.), 85, 86, 106, 111, 458
- \Hfootnotename (hyperref redef. name), 319
- \hfuzz (\hbox warning adjustment), 76
- hline (pack.), 120
- hierarchy of structural commands, 256, 516
- \hline (table com.), 115, 117
- Høgholm, Morten, 209
- \hom (hom math op.), 168, 540
- \hookleftarrow (\hookleftarrow math arrow), 537
- \hookrightarrow (\hookrightarrow math arrow), 537
- horizontal lines (rules), in tabular environments, 115, 117
- horizontal spacing
 - commands
 - _ (interword space), 9, 52, 57, 81, 190, 550
 - \! (negthinspace), 181, 190, 544, 550
 - \, (thinspace), 16, 53, 61, 190, 544, 550
 - \: (medspace), 190, 544, 550
 - \; (thickspace), 190, 544, 550
 - \@. (intersentence space), 550
 - _ (interword space), 550
 - \dotfill (fill com.), 85
 - \frenchspacing, 53
 - \hdotsfor (fill com.), 184, 238, 240
 - \hfill (fill com.), 85, 86, 106, 111, 458
 - \hrulefill (fill com.), 85, 94
 - \hspace, 82, 123
 - \hspace*, 42, 55, 83
 - \medspace, 190, 544, 550
 - \mspace, 190, 544
 - \negmedspace, 190, 544, 550
 - \negthickspace, 190, 544, 550
 - \negthinspace, 181, 190, 544, 550
 - \nobreakspace, 62
 - \nonfrenchspacing, 53
 - \phantom, 82, 189, 191–192, 228, 450
 - \quad, 81, 85, 190, 228, 544, 550
 - \quad, 19, 81, 85, 152, 190, 229, 544, 550
 - \thickspace, 190, 544, 550

- `\thinspace`, 16, 53, 61, 190, 544, 550
 - to avoid using, 525
- `\xspace`, 408–409
 - in math, 14, 152–153, 187–192, 238, 241
 - in text, 9, 42, 50–53, 81–83, 95, 106, 458, 544, 550
- intersentence, 9, 50–53
- interword space (`\`), 9, 50–53, 57, 62, 81, 424, 471
 - preventing removal of, 83
- `howpublished` (BIBTEX database field), 469
- `\href` (hyperref com.), 319
- `\hrulefill` (space fill com.), 85, 94
- `\hslash` (\hbar math sym.), 538
- `\hspace` (spacing com.), 82, 123
- `\hspace*` (spacing com.), 42, 55, 83
- `\Huge` (font size com.), 138, 139, 547
- `\huge` (font size com.), 138, 139, 547
- `\hyperlink` (beamer com.), 339
- hyperlinks, 315
 - in PDF files, 315–319
 - increasing size of, 318
 - suppressing, 318
 - to bibliographic citations, 317
 - to Web sites, 318–319
 - in presentations, 338–342, 353
 - putting bookmarks in documents, 317
 - to bibliographic citations, 317
 - with hyperref pack., 316–318
- `hyperref` (pack.), 316–319
 - bookmarks file (out), 317
 - commands
 - `\autoref`, 316, 318, 319
 - `\pageref`, 318
 - `\ref`, 318
 - `\urladdr`, 319
 - `\writebookmarks`, 318
 - documentation for, 316
 - options, 316
 - `bookmarks=true`, 316
 - `colorlinks`, 316, 317
 - `pagebackref`, 316, 317
- `hyphen.tex`, 68
- hyphenation, 68–71
 - determined by optional hyphen, 12, 68
 - displaying, 70
 - L^AT_EX's algorithm, 65, 68
 - of German text, 145
 - of hyphenated words, 12
 - of words with accents, 68
 - preventing, 69–70, 87
 - rules for English, 71
 - specifying, 68
- `\hyphenation`, 68
- hyphenation exceptions), 71
- hyphens, 11, 48, 61
 - key, 7, 49
 - optional, 12, 68, 76
 - unbreakable, 69
- `\i` (i dotless i), 65, 66, 546
- `\idotsint` ($\int \cdots \int$ large math op.), 159, 171, 542
- `idx` (index aux. file), 509–511, 514
- `\iff` (\iff math arrow), 405, 537
- `\ignorespacesafterend` (spacing com.), 424
- `\iiint` (\iiint large math op.), 159, 171, 542
- `\iint` (\iint large math op.), 159, 171, 542
- `\iint` (\iint large math op.), 159, 171, 542
- `ilg` (*MakeIndex* log files), 511
- illustrations, 27, 264–266
 - double-column, 265
 - `graphicx` (pack.), 254, 265
 - placement of, 82, 267
 - preparation of, 526
 - scaling, 265
 - with `graphicx` (pack.), 27
 - with the `picture` environment, 265
- Illustrator, Adobe, *see under* Adobe Illustrator
- `\Im` (\Im math sym.), 538
- `\imath` (i math sym.), 538
- `\in` (\in binary rel.), 533
- in (inch abs. unit), 12, 76, 85, 87, 92, 447
- `inbibl.tpl` (sample file), 271
- INBOOK (bibl. entry type), 468, 480
- inch (in abs. unit), 12, 76, 85, 87, 92, 447
- `\include` (inclusion com.), 436
 - and counters, 444
- `\includegraphics` (graphics com.), 266
 - file extensions in, 265
 - in presentations, 337, 338
- `\includeonly` (preamble inclusion com.), 444
- including
 - files in other files, 427, 436
 - graphics in documents, 264–266
- INCOLLECTION (bibl. entry type), 468, 475
- incrementing counters, 443
- `ind` (proc. index files), 509–511, 530
- `\indent` (indentation com.), 79
- indentation
 - hanging, 79–80, 125
 - in `multline` and `multline*` environments, 217
 - in source files, 178
 - of environment bodies, 153
 - of first lines of paragraphs, 79, 448
 - of subformulas, 221
 - of text, 79–80
 - customizing, 83
 - forcing, 79

- preventing, 79
- `\index` (indexing com.), 411, 500, 503–509
 - at signs (@) in, 505, 506
 - defining cross-references in indexes with, 507
 - double quote (") in, 507
 - encapsulators, 504
 - exclamation marks (!) in, 503, 504, 507
 - modifiers, 504
 - page ranges in, 504
 - placement of commands, 508, 528
 - spaces in argument of, 511–512
 - syntax of, 508
 - vertical line (l) in, 504, 507
- indexes, 499–514
 - appear in back matter, 500
 - auxiliary files (`idx`), 509, 511
 - cross-references in, 507
 - defining, *see* indexing
 - entries
 - sorting, 505–506
 - stored in `idx` files, 509, 511
 - log files (`ilg`), 511
 - main entries, 274, 499, 503
 - multiple, 513
 - placement of, 500, 511
 - processed files (`ind`), 509, 511, 530
 - processing, 509–511
 - sample, 500–502
 - simple, 274
 - special characters in, 507
 - subentries, 274, 499, 503
 - subsubentries, 274, 499, 504
 - syntax of, 508
 - typesetting, 509–511
 - using `showidx`, 503
 - vertical space in, 274, 499
- indexing, *see also* *MakeIndex*, 499–514
 - commands for, 274, 411, 499–509, 511
 - entries
 - main, 274, 499, 503
 - subentries, 274, 499, 503
 - subsubentries, 274, 499, 504
 - environment for, 274, 451, 499, 509
 - page ranges, 504–505
 - placement of commands for, 508
 - processed files (`ind`), 530
 - processed index files (`ind`), 509, 511
 - rules for, 511–513
 - with AMS document classes, 500
 - with custom commands, 411
- `\indexname` (redef. name), 417
- `\indexspace` (indexing com.), 499
- `\inf` (inf math op.), 168, 308, 540
- information, top matter
 - AMS specific, 287–288
 - AMS subject classifications, 287
 - keywords, 288, 306
- author, 283–287
 - addresses, 284, 305
 - current addresses, 285, 289
 - e-mail addresses, 285, 288, 305
 - Internet addresses, 286, 288, 305
 - research support, 287, 288, 305
- for articles, 281–292
 - author names, 283, 305
 - dedications, 282
 - title, 281, 306
 - translator, 282
- `\infty` (∞ math sym.), 40, 538
- initials
 - periods in, 53
 - typographical rules for, 53
- `\injl` (inj lim math op.), 168
- INKSCAPE, 28
- Inkspace graphics, 28
- inline
 - fractions, 155
 - math formulas, *see* inline math environments
 - tables, 113
- inline math environments, 12, 43, 150
 - `\(` (start math mode), 13, 60, 150
 - `\)` (end math mode), 13, 60, 150
 - act as special braces, 150, 151
 - and \$, 12, 13, 43, 60, 150
 - display-style binomials in, 155
 - font size of, 203
 - math, 150
 - matrices in, 240
 - size of delimiters in, 166
- INPROCEEDINGS (bibl. entry type), 468, 474–475
- `inputenc` (pack.), 145
- `\institute` (beamer com.), 322
- `institution` (BIBTEX database field), 469
- instructions to L^AT_EX, 6, 9, 25, 37, 49, 54, 252, 256
- `\int` (\int large math op.), 16, 171, 542
- integrals, 16, 159, 308
- integration of fonts into L^AT_EX, 143
- interactive mode, 94, 418–419
- `\intercal` (\intercal math op.), 536
- intercolumn spacing, 211, 213, 224
 - in aligned math environments, 228, 233
 - in tabular environments, 116
- interline spacing, 44, 166, 247
 - adjusting, 77–142
- Internet, 286, 318, 460
 - addresses, 286, 319
 - in top matter (`\urladdr` com.), 286, 305, 319

- hyperlinks in PDF documents, 315, 318
 - putting L^AT_EX documents on, 319
- intersentence spaces (`\@.`), 9, 50–53, 550
 - and `\frenchspacing`, 53
 - and `\nonfrenchspacing`, 53
- `\intertext` (text in math com.), 230–231
- interword space, 9, 50–53, 62, 424, 471
 - and `\frenchspacing`, 53
 - and `\nonfrenchspacing`, 53
 - command (`_`), 9, 52, 57, 81, 190, 550
- `intlimits` (opt. of `amsmath` pack.), 308
- invalid characters, 7, 49
- invisible boxes, *see* struts
- invoking
 - custom commands, 406
 - delimited commands, 420
 - proclamations, 37, 105, 108, 338, 425
- `\iota` (ι Greek char.), 162, 532
- ISBN (bibl. com.), 469
- `\it` (obs. L^AT_EX 2.09 font com.), 140
- italic correction, 137–138, 550
 - and commas, 137
 - and `\em`, 138
 - and font commands, 135, 138
 - and `\itshape`, 137, 138
 - and periods, 137
 - and `\slshape`, 138
 - suppressing, 137–138
 - with two-letter font commands, 140
- italics
 - bold math, 200
 - font shape, 11, 20, 54, 104, 133, 135, 136, 139, 456, 546
 - in math mode, 199, 543
 - math font, 199
- `\item` (list item com.), 42–104, 263
 - and square brackets, 102
 - in simple indexes, 274, 499
 - optional arguments of, 100, 102, 453, 454
- `\itemindent` (length com.), 452, 453
- `itemize` (text list env.), 42, 98–100, 263
- `\Itemname` (`hyperref` redef. name), 319
- `\itemsep` (length com.), 451, 457
- `\itshape` (font shape com. dec.), 104, 135, 136, 139, 456, 546
 - and italic correction, 137, 138
- `itshape` (font shape env.), 104
- `\j` (\mathfrak{j} dotless j), 65, 66, 546
- `JabRef`, 497
- Jensen, Frank, 447
- `\jmath` (\mathfrak{j} math sym.), 538
- `\Join` (\bowtie binary rel.), 533
- `journal` (BIB_TE_X database field), 469
- Jr., in bibliographic entries, 470
- justification of text
 - right, 10, 54
- `\k` (ogonek Eur. accent), 549
- `\kappa` (κ Greek char.), 162, 532
- `\ker` (ker math op.), 168, 540
- Kern, Uwe, 321, 344, 560
- key (BIB_TE_X database field), 469
- keyboard, 7–50, 145
- keys, 12
 - alphanumeric, 7
 - Enter, *see* Return key
 - Esc (escape), 49, 405
 - letter, 7–48
 - math symbol, 49
 - math typing, 12
 - modifier, 48, 49
 - number, 48
 - prohibited, 7, 49
 - Return, 7, 9, 49
 - space bar, 7, 9, 49, 321, 330
 - special, 7, 12, 49, 63
 - Tab, 7, 9, 49
- keys, sorting (in `\index` commands), 505–506
 - case sensitivity of, 512
- `\keywords` (top matter com.), 306
 - rules for using, 288
- keywords (bibl. com.), 469
- `\keywordsname` (redef. name), 417
- `\kill` (tabbing com.), 121–123
- Knuth, Donald E., xxii, xxvii, 33, 62, 403, 552, 560
- `koi8-u` (font encoding), 146
- L, slashed (\mathbb{L} , \mathfrak{l}), 67, 545
- `\label` (cross-ref. com.), 20, 23, 39, 72, 153, 219, 235, 256, 259, 263, 324
 - and simple indexes, 274
 - arguments of, 261
 - assigning counters to, 446
 - in table environments, 27, 266
 - placement of commands, 528
 - rules for, 261
 - showing in margin, 262
 - with `\`, 219
- labels
 - for `\bibitem` commands, 271
 - for bibliographic items, 101
 - for equations, 20, 259
 - for groups of equations, 207
 - in list environments, 101
 - of items in a `list` environment, 452–454
 - setting with `\label`, 20, 259
- `\labelsep` (length com.), 452, 453
- `\labelwidth` (length com.), 452, 453

- `\Lambda` (Λ Greek char.), 162, 532
- `\lambda` (λ Greek char.), 162, 532
- Lamport, Leslie, 447, 560
- `\land` (\wedge math op.), 536
- `\langle` (\langle math delimiter), 163, 164, 539
- language (BIBTeX database field), 469
- `\LARGE` (font size com.), 138, 139, 547
- `\Large` (font size com.), 138, 139, 547
- large
 - operators, 16, 18, 159, 169–172, 542
 - limits of, 170, 308
 - sizing of math delimiters with, 166
 - symbols, 97, 231–234
- `\large` (font size com.), 138, 139, 547
- Large Language Model (LLM), 551
- `\larger` (font size com.), 139
- L^AT_EX, 33
 - counters, 443
 - customizing, 403–461
 - distribution, 72
 - components of, 312
 - directories, 262
 - document classes, *see under* document classes
 - document structure, 251–278, 517
 - documents, 251–278
 - putting on the Internet, 319
 - font substitution, 139, 527
 - implementations, 6, 34, 193
 - UNIX, 511
 - localization of, 65, 145–146
 - numbers stored by, 66–68
 - release dates of, 72, 254
 - source files, *see under* source files
 - spacing in text, 9
 - structure of, 33–34
 - updates to, 194
 - using, 34, 35
 - versions of
 - specifying, in documents, 72, 254
 - writing books with, 515–530
- `\LaTeX` (L^AT_EX logo com.), 6, 66
- L^AT_EX 2.09, 427
 - two-letter font commands, 139, 140
- L^AT_EX 2_ε, *see* L^AT_EX
- L^AT_EX packages, 34, 56, 254
- `\LaTeX2e` (L^AT_EX2e logo com.), 66
- latexsym (pack.), 178, 253, 303, 309
- layout
 - of a list, 452
 - of Computer Modern typewriter font, 64, 65
 - of pages, 33, 275–278
- `\layout` (page-layout diagram com.), 275
- layout (pack.), 275
- `\lbrace` ($\{$ math delimiter), 43, 163, 539
- `\lbrack` ($[$ math delimiter), 163, 539
- LC3, xxii, xxv, xxvii, xxxi, 34, 74, 80, 81, 93, 94, 104, 111, 120, 123, 127, 143–145, 258, 262, 267, 268, 277, 319, 447, 458, 490, 495, 513, 514, 520, 560
- `\lceil` (\lceil math delimiter), 163, 539
- `\ldots` (...)
 - in math, 43, 158, 185
 - in text, 16
- `\le` (\leq binary rel.), 533
- `\leadsto` (\leadsto math arrow), 537
- `\left` (math delim.), 43, 164–166, 183, 240
 - blank, 164, 181
 - must be balanced, 164, 221
- left double quote (“)
 - text symbol, 61, 67, 548
 - typing, 9
- left single quote (‘)
 - key, 7, 9, 48, 61
 - text symbol, 61, 549
- `\left(` ($($ math delimiter), 16, 18, 43, 163, 164, 183, 221, 240
- `\left.` (blank math delim.), 164, 181
- `\left<` (\langle math delimiter), 164
- `\left[` ($[$ math delimiter), 16, 163, 180
- `\Leftarrow` (\Leftarrow math arrow), 537
- `\leftarrow` (\leftarrow math arrow), 404, 414, 537
- `\leftarrowtail` (\leftarrowtail math arrow), 537
- `\leftharpoondown` (\leftharpoondown math arrow), 537
- `\leftharpoonup` (\leftharpoonup math arrow), 537
- `\leftleftarrows` (\leftleftarrows math arrow), 537
- `\leftmargin` (length com.), 451, 452, 455
- `\Leftrightarrow` (\Leftrightarrow math arrow), 537
- `\leftrightarrow` (\leftrightarrow math arrow), 537
- `\leftrightharpoons` (\leftrightharpoons math arrow), 537
- `\leftrightsquigarrow` (\leftrightsquigarrow math arrow), 537
- `\leftroot` (root-adjustment com.), 160
- `\leftthreetimes` (\leftthreetimes math op.), 536
- legacy
 - document classes, 98, 253, 258, 311, 312
 - environments, 123–125, 451
 - quotation, 124
 - quote, 123
 - verse, 124
- legalpaper (doc. class opt.), 71, 307
- Lehman, Philipp, 560
- length commands, 87, 88, 277, 443, 447–450
 - and boxes, 449
 - arithmetic with, 449
 - `\baselineskip`, 80, 142
 - `\baselinestretch`, 78
 - defining new, 448
 - `\depth`, 87, 88, 92
 - `\fboxrule`, 91
 - `\fboxsep`, 91

- `\hangafter`, 79
- `\hangindent`, 79
- `\height`, 87, 88, 92
- in list environments, 451–453
- `\itemindent`, 452, 453
- `\itemsep`, 451, 457
- `\labelsep`, 452, 453
- `\labelwidth`, 452, 453
- `\leftmargin`, 451, 452, 455
- `\listparindent`, 452, 453, 458
- `\marginparpush`, 448
- `\marginparwidth`, 94
- `\medskipamount`, 418, 450
- `\multlinegap`, 217
- `\oddsidemargin`, 277
- `\overfullrule`, 75
- `\parindent`, 448, 459
- `\parsep`, 451, 452, 457, 458
- `\parskip`, 448, 451, 452, 459, 525
- `\partopsep`, 451
- `\rightmargin`, 451, 452, 455
- setting, 79, 80, 94, 448–450
- `\textwidth`, 277, 443, 448
- `\topsep`, 451, 452
- `\totalheight`, 87, 88, 91, 92
- `\width`, 87, 88, 92
- lengths, rubber, *see* rubber lengths
- `\leq` (\leq binary rel.), 533
- `\leqno` (doc. class opt.), 307
- `\leqq` (\leq binary rel.), 534
- `\leqslant` (\leq binary rel.), 534
- less than ($<$)
 - as binary relation, 533
 - text symbol, 67, 548
- `\lessapprox` (\lesssim binary rel.), 534
- `\lessdot` (\lessdot binary rel.), 534
- `\lesseqgtr` (\lesseqgtr binary rel.), 534
- `\lesseqqgtr` (\lesseqqgtr binary rel.), 534
- `\lessgtr` (\lessgtr binary rel.), 534
- `\lesssim` (\lesssim binary rel.), 534
- `letter` (doc. class), 312
- `letter` (letter env.), 312
- letter keys, 7–48
- `letterpaper` (doc. class opt.), 307
- letters
 - accented, 11
 - counter style, 445
 - Greek, 162, 180, 430–432, 532
 - Hebrew, 162, 531
- Levy, Silvio, 561
- `\lfloor` (\lfloor math delimiter), 163, 539
- `\lg` (\lg math op.), 168, 540
- `\lhd` (\lhd math op.), 536
- ligatures, 64
 - suppressing, 64, 179, 405
- `\lim` (\lim math op.), 168, 308, 540
- `\liminf` (\liminf math op.), 168, 540
- limits
 - as subscripts, 16, 159, 167, 169, 170, 180, 181, 307, 416
 - as superscripts, 16, 159, 170, 180, 307
 - large operators with, 16, 18, 159, 169–171, 308, 542
 - multiline, 170–172
 - operators with, 167, 168, 308, 540
 - placement of, 308
- `\limits` (limit-control com.), 159, 170
- `\limsup` (\limsup math op.), 168, 540
- line boxes, *see under* boxes
- line ending characters, 49–51, 54, 71, 81
 - commenting out, 71, 72, 508
 - translation of, 49
- Line numbers for PDF files, 320
- line numbers, in error messages, 151, 152
- `\linebreak` (line breaking com.), 527, 528
- `\linebreak` (line-breaking com.), 76, 77
 - optional arguments of, 78
- `\linenumbers`, 320
- lines
 - adding to
 - lists of figures and tables, 522
 - tables of contents, 519–520, 530
 - blank
 - in bibliographies, 317
 - in displayed math environments, 39, 151, 154, 219
 - in inline math environments, 151
 - in text environments, 43, 98
 - in top matter commands, 281
 - in verbatim environments, 126
 - terminating paragraphs with, 9, 43, 50, 79, 98, 124, 317
 - breaking, 76
 - preventing, 78
 - with `\`, 17, 21, 23, 25, 77, 104, 115, 121, 123, 125, 172, 215, 216, 219, 220, 231, 235, 247, 281–285, 306, 353
 - distance between, 84
 - adjusting, 95
 - filling with space or dots, 85–86, 94, 106, 184, 238, 240, 458
 - separating
 - with `\`, 17, 21, 23, 25, 77, 104, 115, 121, 123, 125, 172, 215, 216, 219, 220, 231, 235, 247, 281–285, 306, 353
 - with `\linebreak`, 76, 77, 527, 528
 - stretchable, 175
 - too wide warnings, 11
 - wrapping, 70
- lines (rules)

- horizontal
 - filling lines with, [85](#), [94](#)
 - in tabular environments, [115](#), [117](#)
- intersecting in tables, [120](#)
- vertical
 - in CD environments, [246](#)
 - in tabular environments, [115](#)
- links, hyper, *see* hyperlinks
- `\list` (list text env.), [42](#), [98](#), [451–458](#)
 - arguments of, [454](#)
 - length commands in, [451–453](#)
 - using counters with, [455](#)
- list text environments, [42](#), [98–104](#)
 - description, [100](#)
 - `\enumerate`, [98](#), [103](#)
 - in proclamations, [106](#)
 - `\itemize`, [42](#), [99](#)
 - `\list`, [42](#), [98](#)
 - mixing, [102](#)
 - nesting, [101](#)
 - rules for, [100](#)
 - `\trivlist`, [458](#)
- `\listfigurename` (redef. name), [417](#)
- `\listfiles` (file list com.), [428](#)
- listing files used by a document, [428](#)
- `\listoffigures` (front matter com.), [522](#)
- `\listoftables` (front matter com.), [521](#)
- `\listparindent` (length com.), [452](#), [453](#), [458](#)
- lists
 - cross-referencing items in, [99](#), [103](#)
 - custom
 - and counters, [454](#)
 - defining, [451–459](#)
 - environments for, *see* list text environments
 - in presentations, [335–336](#)
 - of figures, [521–522](#)
 - adding a line to, [522](#)
 - auxiliary file (`\lof`), [522](#), [529](#)
 - fragile commands in, [60](#)
 - of tables, [114](#), [521–522](#)
 - adding a line to, [522](#)
 - auxiliary file (`\lot`), [521](#), [529](#)
 - fragile commands in, [60](#)
- `\listtablename` (redef. name), [417](#)
- `\ll` (\ll binary rel.), [533](#)
- `\llcorner` (\llcorner math delimiter), [163](#), [539](#)
- `\Lleftarrow` (\Leftarrow math arrow), [537](#)
- `\lll` (\lll binary rel.), [534](#)
- `\ln` (\ln math op.), [168](#), [540](#)
- `\lnapprox` (\lesssim neg. binary rel.), [535](#)
- `\lneq` (\lesseqgtr neg. binary rel.), [535](#)
- `\lneqq` (\lesseqgtr neg. binary rel.), [535](#)
- `\lnot` (\neg math sym.), [538](#)
- `\lnsim` (\lesssim neg. binary rel.), [535](#)
- local commands, *see under* scope
- localization, of L^AT_EX, [145–146](#)
- locating errors, [74](#)
- location of BibT_EX database and style files, [483](#)
- `\lof` (list of figures file), [522](#), [529](#)
- `\log` (log math op.), [168](#), [540](#)
- `\log` (L^AT_EX log file), [6](#), [20](#), [36](#), [70](#), [75](#), [419](#), [427](#), [521](#)
 - error and warning messages recorded in, [12](#), [75](#)
- log files
 - for BibT_EX, *see* `\blg`
 - for indexes, *see* `\ilg`
 - for L^AT_EX, *see* `\log`
- log-like functions, *see* operators
- logical
 - design, [37](#), [251](#)
 - of books, [522–525](#)
 - units, [9](#), [25–259](#), [268](#), [274](#), [292](#), [451](#), [481](#), [499](#), [509](#), [516–517](#)
 - hierarchy of, [256](#)
 - numbering of, [447](#), [516](#)
 - numbering of proclamations within, [108](#)
- logos, [66–68](#)
- long commands, [60](#)
- `\Longleftarrow` (\Longleftarrow math arrow), [537](#)
- `\longleftarrow` (\leftarrow math arrow), [537](#)
- `\Longleftrightarrow` (\Longleftrightarrow math arrow), [537](#)
- `\longleftrightarrow` (\longleftrightarrow math arrow), [537](#)
- `\longmapsto` (\mapsto math arrow), [537](#)
- `\Longrightarrow` (\Longrightarrow math arrow), [537](#)
- `\longrightarrow` (\longrightarrow math arrow), [537](#)
- `\looparrowleft` (\looparrowleft math arrow), [537](#)
- `\looparrowright` (\looparrowright math arrow), [537](#)
- `\lor` (\vee math op.), [536](#)
- `\lot` (list of tables files), [521](#), [529](#)
- low-level font commands, [142](#)
- lowercase counter styles
 - letters (`\alph`), [445](#)
 - roman numerals (`\roman`), [103](#), [445](#)
- lowline (`_` text symbol), [63](#), [67](#), [548](#)
- `\lozenge` (\Diamond math sym.), [538](#)
- `\lrcorner` (\lrcorner math delimiter), [163](#), [539](#)
- `\Lsh` (\Lsh math arrow), [537](#)
- `\ltimes` (\ltimes math op.), [536](#)
- Lucida Bright, [143–145](#)
- lucidabr (pack.), [145](#)
- `lucidabr.dtx` (PSNFSS distr. file), [144](#)
- `lucidabr.fdd` (PSNFSS distr. file), [144](#)
- `lucidabr.ins` (PSNFSS distr. file), [144](#)
- `lucidabr.sty` (PSNFSS distr. file), [145](#)
- `lucidabr.yy` (PSNFSS distr. file), [144](#)
- `\vertneqq` (\neq neg. binary rel.), [535](#)
- LY1 (pack. opt.), [145](#)

- Mac, *see* Macintosh
- Macintosh, 5
- macron (˘ text accent), 66, 546
- MacTeX (Windows L^AT_EX front end), 5
- main entries, in indexes, 274, 499, 500, 503, 506, 507
- main matter, 253, 255–268, 517
- \mainmatter (struct. com.), 517
- \makebox (box com.), 86–88, 90, 92, 94
- makeglos (pack.), 514
- \makeglossary (preamble glossary com.), 514
- makeidx (pack.), 500
- MakeIndex, 64, 275, 499–514
 - and font commands, 136
- \maketitle (title-page com.), 252, 253, 518
 - and abstracts, 255, 292
 - and page styles, 277
- MANUAL (bibl. entry type), 468, 480
- manuals, in B^IB^TE^X database files, 480
- manuscripts
 - in B^IB^TE^X database files, 479–480
 - preparing for publication, 515
- \mapsto (\mapsto math arrow), 537
- \mapstochar (\mapsto math arrow), 537
- marginal comments, 93–94, 527
 - in double-sided documents, 93
 - in equations, 93
 - in multiline math environments, 93
 - space between, 448
 - width of paragraph box for, 94
- \marginpar (marginal comment com.), 93
 - in double-sided documents, 93
- \marginparpush (length com.), 448
- \marginparwidth (length com.), 94
- margins, 448, 451
 - of nested lists, 451
 - showing symbolic references in, 262
- \markboth (left and right page header com.), 277
 - arguments of, 277
- \markleft (left page header com.), 277, 289
- \markright (right page header com.), 277
- MASTERSTHESIS (bibl. entry type), 469, 477–478
- math, 6
 - accents, 17, 173, 182, 183, 310, 418, 526
 - alphabets, 179–181, 184, 199–201, 310, 433, 543
 - arrows, 175, 187, 245, 537
 - as delimiters, 163, 164
 - stretchable, 175, 245–247
 - vertical, 246
 - binary operations, *see* binary operations
 - binary relations, *see* binary relations
 - characters, 12, 49
 - environments, 150–152
 - aligned, *see under* aligned, *see under* aligned
 - and \verb commands, 128
 - blank lines in, 151, 154
 - displayed, *see* displayed math environments, *see* displayed math environments
 - inline, *see* inline math environments
 - multiline, 219–223, 330
 - page breaks in, 247
 - spaces in, 151
 - fonts, 199–201
 - blackboard bold, 180, 201, 433, 543
 - bold, 182, 199, 201, 433, 543
 - bold italic, 200
 - calligraphic, 184, 200, 310, 433, 543
 - Euler Fraktur, 179, 201, 310, 433, 543
 - Euler Script, 310, 433, 543
 - italic, 199, 543
 - roman, 199, 543
 - sans serif, 199, 543
 - size of, 203
 - typewriter, 199, 543
 - mode, *see* math mode
 - multiline, 211–248
 - Visual Guide* to, 211, 212
 - operators, 15, 16, 18, 41, 155, 167–172, 412, 536, 540
 - declaring, 42, 415–416
 - large, 16, 18, 159, 166, 169–172, 308, 542
 - with limits, 16, 18, 159, 167, 168, 170, 171, 308, 540, 542
 - subscripts and superscripts, 203
 - symbol alphabets, 200–201
 - blackboard bold, 201
 - calligraphic, 200
 - Euler Fraktur, 201
 - Greek, 200
 - symbols, *see* math symbols
 - text in, 14, 19, 69, 78, 86, 87, 152, 160–161, 179, 200, 230–231, 246, 310
 - specifying fonts for, 161
 - typing, 25, 209
- math (inline math env.), 150
- math arrows, 175, 187, 245
 - as delimiters, 163, 164, 539
 - stretchable, 175, 245–247
 - vertical, 246
- math commands, 15, 18, 157, 159, 170, 191, 197, 203, 207
 - for alignment of formulas, 218
- math delimiters, 16, 18, 163–167, 179–181, 183, 184, 188, 189, 191, 221, 240, 539
 - and large operators, 166
 - arrows, 163

- balancing, 164, 181, 221
- blank, 164, 181
- fixed-size, 165
- in inline math environments, 166
- in `smallmatrix` subsidiary math environments, 240
- left bracket (`[`), 16, 163, 180
- limitations of stretching, 165
- right bracket (`]`), 16, 163, 180
- specifying size of, 165, 183
- stretchable, 43, 164–165
- with matrix variants, 239–240
- math error messages, 40, 41
- math font commands, *see also* font command declarations *and* font commands
 - for bold
 - `\boldsymbol`, 182, 200, 201, 310, 543
 - `\mathbb`, 180, 201, 543
 - `\mathbf`, 182, 199, 201, 543
 - `\pmb`, 202, 310
 - for italics
 - `\mathit`, 199, 543
 - for series
 - `\mathnormal`, 199, 200, 543
 - for shape
 - `\mathcal`, 184, 200, 543
 - `\mathfrak`, 179, 201, 310, 543
 - `\mathrm`, 199, 543
 - `\mathscr`, 310, 543
 - `\mathsf`, 199, 543
 - `\mathtt`, 199, 543
 - for size
 - `\displaystyle`, 203, 207
 - `\scriptscriptstyle`, 203, 207
 - `\scriptstyle`, 203, 207
 - `\textstyle`, 203, 207
 - for weight
 - `\boldsymbol`, 182, 200, 201, 310, 543
 - `\mathbf`, 182, 199, 201, 543
 - `\pmb`, 202, 310
- math mode, 6, 48, 62, 128, 152, 160, 167, 202, 409, 415, 416, 436
- math symbols, 14, 40, 49, 111, 173, 181, 245, 253, 303, 161–544
 - alphabets, 200–201
 - and delimiters, 16, 163, 164
 - bold, 201–202, 310
 - building new, 181, 194–198, 412
 - classification of, 188–189, 199
 - declaring types of, 198
 - end of proof, 94, 415
 - in text, 405
 - large, 231–233
 - negated, 197–198
 - shorthand commands for, 404
 - side-setting, 197–198
 - sizes of, 203
 - spacing of, 187–192
 - stacking, 44, 194–197
 - STIX, 187–194
 - stretchable, 173–175
 - suppressing, 111, 112
- math units, *see* `\mu`
- `\mathbb` (\mathbb{X}), 180, 201, 543
- `\mathbf` (math font weight com.), 182, 199, 201, 543
- `\mathbin` (binary-op. com.), 198
- `\mathcal` (\mathcal{X}), 184, 200, 543
 - and Euler Script, 310
- `\mathfrak` (\mathfrak{X}), 179, 201, 310, 543
- `\mathit` (math font shape com.), 199, 543
- `\mathnormal` (math font shape com.), 199, 200, 543
- `\mathrel` (binary-rel. def. com.), 198
- `\mathring` (\mathring{x} math accent), 172, 543
- `\mathrm` (math font shape com.), 199, 543
- `\mathscr` (opt. of `eucal` pack.), 310, 427, 433, 543
- `\mathscr` (\mathcal{X} math font shape com.), 310, 543
- `\mathsf` (math font shape com.), 199, 543
- `\mathstrut` (spacing com.), 95, 204
- `mathtools` (pack.), 209
- `\mathtt` (math font shape com.), 199, 543
- `\mathversion`, 201–202
- matrices, 17–18, 237–243
 - in inline math environments, 240
- matrix (subsid. math env.), 17–18, 212, 214, 236–243
 - variants, 239–240
- `\max` (max math op.), 168, 308, 540
- `MaxMatrixCols` (counter), 238
- `\mbox` (box com.), 69, 86, 87
- McLean, Ruari, 560
- McPherson, Kent, 275
- `\mdseries` (font weight com. dec.), 135, 138, 546
- `\measuredangle` (\angle math sym.), 538
- measurements, of text boxes, 449
 - commands for, 450
- medium (font weight), 134, 135, 138, 546
- `\medskip` (spacing com.), 84
- `\medskipamount` (length com.), 418, 450
- `\medspace` (spacing com.), 190, 544, 550
- Mendeleev, 497
- messages, 6, 37–41
 - Argument of `\xxx` has an extra `]`, 126
 - Bad math environment delimiter, 112, 151, 521

- `\begin{document}` ended by `\end{xxx}`,
73, 126
- `\begin{split}` not allowed here, 236
- `\begin{xxx}` on input line `xx`
ended by `\end{yyy}`, 38, 59
- Can be used only in preamble, 254
- Command `\xxx` already defined, 407, 418,
422
- Display math should end with `$$`, 39
- Double superscript, 156
- Environment `xxx` undefined, 422
- error, *see* messages
- Extra alignment tag has been
changed to `\cr`, 242
- Extra `}`, or forgotten `\xxx`, 227
- I was expecting a `'`,
or a `''`, 488, 489
- Illegal character in array arg, 241
- Illegal unit of measure (pt
inserted), 95
- Invalid use of `\xxx`, 160
- line numbers in, 151, 152
- Misplaced `\xxx`, 115, 231
- Missing control sequence
inserted, 448
- Missing `$` inserted, 39, 40, 128, 151,
152, 154, 238
- Missing number, treated
as zero, 78, 92
- Missing `}` inserted, 40, 227
- No counter `'xxx'` defined, 444
- Paragraph ended before `\xxx`
was complete, 38, 95, 106, 242, 413,
420, 512
- recorded in log file, 36
- Runaway argument?, 38, 106, 242, 292,
420, 512
- Runaway definition?, 39
- showing paragraph breaks in, 79
- Something's wrong--perhaps
a missing `\item`, 100, 272
- `\tag` not allowed here, 219
- Text line contains an invalid
character, 40, 50
- Too many `}`'s, 58, 292
- Undefined control sequence, 38, 57,
290, 292, 407
- Underfull `\hbox`, 77
- Use of `\xxx` doesn't match its
definition, 420
- `\verb` command ended by end of line,
128
- warning, *see* warning messages
- with L^AT_EX, 487–489
- You're missing a field name, 488
- `\mho` (U math sym.), 538
- Microsoft
typography web page, 143
- `\mid` (| binary rel.), 43, 167, 180, 189, 533
- minus, 7, 15, 49, 62, 188, 191, 192, 199, 220,
222
- as binary operation, 536
- MISC (bibl. entry type), 469, 480
- Mittelbach, Frank, 81, 559, 560
- mixing list text environments, 102
- `\mod` (mod math op.), 169
- `\models` (\models binary rel.), 533
- modes
 - interactive, 94, 418–419
 - typographic
 - math, *see also* inline and displayed
math environments, 6, 48, 62, 128, 152,
160, 167, 202, 409, 415, 416, 436
 - text, 6, 48, 49, 152, 179, 409, 415, 436
- modifier keys, 48, 49
- modifiers, in `\index` commands, 504
combining, 504
- modifying environments, 422–424
- monospaced fonts, 133, 134
- `\month` (time com.), 66
- month (L^AT_EX database field), 469, 480
- Moore, Ross, 247, 559
- movable arguments of commands, 60, 521
- `\mp` (\mp math op.), 536
- `mpfootnote` (counter), 443
- `mrabbrev.bib` (L^AT_EX database file), 480
- `mrnumber` (bibl. com.), 469
- `\mspace` (spacing com.), 190, 544
- `\mu` (μ Greek char.), 162, 532
- `mu` (math unit, rel.), 190, 544
- `multicol` (pack.), 81
- `multicols` (disp. text env.), 81
- `\multicolumn` (table com.), 117–119
- multicolumn text
 - in documents, 81
 - in tables, 117–119
- multiline
 - boxes, 86
 - formulas, 21–24, 43, 211–248
 - Visual Guide* to, 211, 212
 - aligning, 221–222
 - splitting, 216–218
- limits, 170–172
- math environments, 330
 - adjusting columns in, 211, 214, 236–244
 - aligning, 211, 214
 - marginal comments in, 93
 - page breaks in, 247–248
- subscripts and superscripts, 170–172
- table entries, 117, 118

- `\multimap` (\multimap math arrow), 537
- multiple
 - arguments
 - in custom commands, 410, 412
 - authors
 - in bibliographies, 470
 - in documents, 288–289
 - bibliographies in a document, 274
 - captions, 266
 - citations, 273
 - indexes, 513
 - spaces, 14, 50, 57
 - tables of contents, 520
- multiplication, 15, 155, 158
- `\multline` (disp. math env.), 212, 214, 216–218
 - indentation of lines in, 217
 - rules for, 216
- `\multline*` (disp. math env.), 217–218
 - indentation of lines in, 217
- `\multlinegap` (length com.), 217
- `myams.tpl`, 303–306
- `myheadings` (page style), 277
- n -th root, 18, 159
- `\nabla` (∇ math sym.), 538
- `\namelimits` (opt. of `amsmath` pack.), 308
- names
 - for abstracts, 255
 - for proclamations, 106, 108
 - of authors
 - in articles, 283, 305
 - in bibliographies, 470
 - in running heads, 283
 - of commands, 54, 55, 427
 - of counters, 108
 - redefinable, 145, 274, 318, 319, 403, 416–417
 - tagging equations with, 21, 205–207, 215, 217, 219, 222, 234, 235
- `\natural` (\natural math sym.), 538
- `\ncong` (\ncong neg. binary rel.), 535
- `\ne` (\neq neg. binary rel.), 197, 535
- `\nearrow` (\nearrow math arrow), 537
- `\NeedsTeXFormat` (preamble com.), 72, 254
- `\neg` (\neg math sym.), 538
- negated math symbols, 197–198, 535
- `\negmedspace` (spacing com.), 190, 544, 550
- `\negthickspace` (spacing com.), 190, 544, 550
- `\negthinspace` (spacing com.), 181, 190, 544, 550
- `\neq` (\neq neg. binary rel.), 535
- nesting
 - of comment environments, 73
 - of list text environments, 101
- `\newcommand` (custom com.), 41, 61, 404–421
 - defining arguments with, 410
 - name already in use, 418
 - optional arguments, 414
- `\newcommand*` (custom com.), 413
- `\newcounter` (custom counter com.), 444, 448
 - optional argument of, 444
- `\newenvironment` (custom env.), 422–424
- `\newenvironment*` (custom env.), 426
- `\newlabel` (aux. file com.), 36
- `newlattice` (pack.), 254, 427, 428
- `newlattice.sty` (sample file), 421, 424, 427–436
- `\newlength` (length-command def. com.), 448
- `\newline` (line breaking com.), *see also* `\`, 76
- `\newline` (line-breaking com.), 76, 78
- `\newpage` (page breaking com.), 80, 268
- `\newtheorem` (procl. com.), 105, 107, 109, 111, 303, 322, 446
 - optional arguments of, 107, 108
 - syntax of, 108
- `\newtheorem*` (procl. com.), 109, 423
- `\newtheoremstyle` (procl. com.), 111
- `\nexists` (\nexists math sym.), 538
- `\NG` (Eng Eur. char.), 549
- `\ng` (eng Eur. char.), 549
- `\ngeq` (\ngeq neg. binary rel.), 535
- `\ngeqq` (\ngeqq neg. binary rel.), 535
- `\ngeqslant` (\ngeqslant neg. binary rel.), 535
- `\ngtr` (\ngtr neg. binary rel.), 535
- `\ni` (\ni binary rel.), 533
- `\nLeftarrow` (\nLeftarrow math arrow), 537
- `\nleftarrow` (\nleftarrow math arrow), 537
- `\nLeftrightarrow` (\nLeftrightarrow math arrow), 537
- `\nlefttriarow` (\nlefttriarow math arrow), 537
- `\nleq` (\nleq neg. binary rel.), 535
- `\nleqq` (\nleqq neg. binary rel.), 535
- `\nleqslant` (\nleqslant neg. binary rel.), 535
- `\nless` (\nless neg. binary rel.), 535
- `\nmid` (\nmid neg. binary rel.), 197, 535
- `noamsfonts` (doc. class opt.), 309
- `\nobreakdash` (hyph. prev. com.), 69
- `\nobreakspace` (spacing com.), 62
- `\nocite` (bibl. com.), 483
- `\nocite*` (bibl. com.), 483
- `\coordinate`, `TikZ` command, 376
- `\nofiles` (preamble com.), 529
 - and auxiliary files, 529
- `\noindent` (indentation-suppression com.), 79, 527
- `\nointlimits` (opt. of `amsmath` pack.), 308
- `\nolimits` (limit-control com.), 159, 169, 170, 181
- `\nolinebreak` (linebreak-suppression com.), 78
- `nomath` (doc. class opt.), 309

- nonamellimits (opt. of amsmath pack.), 308
- nonbreakable spaces (~ tie), 9, 49, 52, 62, 78, 550
 - absorb spaces, 63
 - in BibTeX databases, 471
 - with cross-references, 262
- \nonfrenchspacing (spacing com.), 53
- \nopagebreak (page break suppression com.), 80
- normal
 - document font family, 134
 - font shape
 - command declarations for, 103, 135, 144, 546
 - commands for, 135, 144, 546
 - math commands for, 161, 199, 200, 543
- \normalfont (font shape com. dec.), 103, 135, 144, 161, 546
- \normalsize (font size com.), 138, 139, 547
- nosumlimits (opt. of amsmath pack.), 308
- \not (math com.), 197
- \notag, 22, 23, 215, 217, 219, 222, 235
- notation (text env.), 323
- notcite (opt. of showkeys pack.), 262
- note (beamer doc. class opt.), 350
- \note (beamer com.), 350
- note (BibTeX database field), 469
- notes, in presentations, 350, 351
- \notin (\notin neg. binary rel.), 197, 535
- notitlepage (doc. class opt.), 308, 518
- \nparallel (\nparallel neg. binary rel.), 535
- \nprec (\nprec neg. binary rel.), 535
- \npreceq (\npreceq neg. binary rel.), 535
- \nrightarrow (\nrightarrow math arrow), 537
- \nrightarrow (\nrightarrow math arrow), 537
- \nshortmid (\nshortmid neg. binary rel.), 535
- \nshortparallel (\nshortparallel neg. binary rel.), 535
- \nsim (\nsim neg. binary rel.), 535
- \nsubseteq (\nsubseteq neg. binary rel.), 180, 535
- \nsubseteqq (\nsubseteqq neg. binary rel.), 535
- \nsucc (\nsucc neg. binary rel.), 535
- \nsucceq (\nsucceq neg. binary rel.), 535
- \nsupseteq (\nsupseteq neg. binary rel.), 535
- \nsupseteqq (\nsupseteqq neg. binary rel.), 535
- \ntriangleleft (\ntriangleleft neg. binary rel.), 535
- \ntrianglelefteq (\ntrianglelefteq neg. binary rel.), 535
- \ntriangleright (\ntriangleright neg. binary rel.), 535
- \ntrianglerighteq (\ntrianglerighteq neg. binary rel.), 535
- \nu (ν Greek char.), 162, 532
- number (BibTeX database field), 468, 469
- numbered lists, *see* enumerate
- numbering
 - automatic, 19, 21, 36, 99, 106, 235, 256
 - of appendices, 259, 517
 - of equations, 19–21, 42, 153–154, 205, 215, 217, 219
 - groups, 222–223
 - in chapters of books, 517
 - in multiline math environments, 219
 - is relative, 205, 206
 - of subsidiary math environments, 234, 235
 - suppressing, 22, 23, 154, 205, 215, 217, 219, 222, 235
 - variations, 206, 222
 - within sections, 19, 154, 254
 - within split subsid. math align. env., 234–235
- of figures, 264
- of pages, style of, 445
- of proclamations, 106–108
 - consecutively, 107–108
 - suppressing, 109, 110, 423
 - within sections, 107
- of structural units, 444, 516
 - controlling, 447
 - in books, 516
 - sections, 256
 - suppressing, 257, 517
- of tables, 264
- renumbering, 20, 36
- with counters, 443–450
 - resetting, 444, 446
- \numberline (toc file com.), 521
- numbers
 - counter styles for, 103, 445–446, 455
 - for equations, placement of, 307
 - for proclamations, position of, 111
 - in bibliographic fields, 468
 - in lists, 99
 - keys, 48
 - line
 - in error messages, 151, 152
 - in warning messages, 12
 - page
 - referencing, 264
 - style of, 445
 - ranges, 11, 62, 67
 - in BibTeX databases, 468, 471
 - in index entries, 504
 - real, in length commands, 447
 - stored by L^AT_EX, 66–68
 - stored in
 - counters, 443
 - registers, 419
- \numberwithin, 19, 154, 254, 516
- numeric counter style (\arabic), 445, 455
- \nVDash (\nVDash neg. binary rel.), 535
- \nvDash (\nvDash neg. binary rel.), 535
- \nvDash (\nvDash neg. binary rel.), 535

- `\nvdash` (\nmid neg. binary rel.), 535
- `\nwarrow` (\nwarrow math arrow), 537
- O'Sean, Arlene Ann, 221, 561
- O, slashed (\emptyset , \varnothing), 67, 545
- obtaining
 - sample files for this book, 5
- octothorp (#), 63, 67, 548
- `\oddsidemargin` (length com.), 277
- `\odot` (\odot math op.), 536
- `\OE` ethel (Æ), 67, 545
- `\oe` ethel (æ), 67, 545
- `\ogonek` (Eur. accent), 549
- `\oint` (\oint large math op.), 159, 171, 542
- old-style digits, 48
- `\oldstylenums` (old-style digits com.), 48
- `\Omega` (Ω Greek char.), 162, 532
- `\omega` (ω Greek char.), 162, 532
- `\ominus` (\ominus math op.), 536
- `onecolumn` (doc. class opt.), 308
- `\onecolumn` (single-column com.), 81
- `\onehalfspacing` (setspace com.), 79
- `oneside` (doc. class opt.), 308, 518
- `\only` (beamer com.), 327, 330–332, 334–335, 338
- `\onslide` (beamer com.), 327, 331, 332, 334–335, 338
- OpenAI, 552, 554
- `openany` (doc. class opt.), 518
- `openright` (doc. class opt.), 518
- operations
 - arithmetical, 15, 155–157
 - binary, *see* binary operations
- operators, math, 15, 16, 18, 41, 155, 167–172, 412, 536, 540
 - declaring, 42, 415–416
 - large, 16, 18, 159, 166, 169–172, 308, 542
 - sizing of math delimiters with, 166
 - with limits, 16, 18, 159, 167, 168, 170, 171, 308, 540, 542
 - defining, 198, 200, 416
 - without limits
 - defining, 198, 200, 202, 254, 310, 415
- `\oplus` (\oplus math op.), 536
- optional
 - arguments, 57, 78, 80, 93, 94, 96, 106, 159, 160, 175, 238, 247, 254
 - multiple, 56
 - of `\`, 76, 78, 247
 - of commands, 100, 102, 103, 107, 203, 205, 273, 284–287, 347, 444, 446, 451, 453, 454, 458, 522
 - of custom commands, 414
 - of custom environments, 421, 424–426
 - of environments, 106, 108, 112, 115, 233, 248, 267, 425, 529
 - of sectioning commands, 257, 519
 - of structural commands, 259, 516
 - of top matter commands, 281–289, 327
 - use square brackets (`[]`), 18, 55–56, 103, 112, 115, 292
- bibliographic fields, 470, 472–480
- hyphens, 12, 68, 76
- options
 - of packages, *see* package options
- organization (BIBTeX database field), 469, 475
- orthogonality of font commands, 139–140
- `\oslash` (\oslash math op.), 536
- OT1 (Old Text encoding), 141
- `\otimes` (\otimes math op.), 536
- out (hyperref bookmarks file), 317
- `\overbrace` (horizontal brace com.), 173, 182
 - with a superscript, 174
- `\overdot` ($\dot{}$ text accent), 66, 546
- `\overfullrule` (length com.), 75
- overlapping braces, 59
- overlays, *see also under* presentations
 - commands for, 334
 - layering, 330
- `\overleftarrow` ($\overleftarrow{}$ math arrow), 175
- `\overleftrightarrow` ($\overleftrightarrow{}$ math arrow), 175
- `\overline` ($\overline{}$ math line sym.), 173, 175, 183
- overprint (beamer env.), 354
- `\overrightarrow` ($\overrightarrow{}$ math arrow), 175
- `\overset` (symbol-building com.), 181, 194, 412
 - creating binary operations and relations with, 196
- `\owns` (\owns binary rel.), 533
- `\P` (\P pilcrow or paragraph)
 - math symbol, 538
 - text symbol, 67, 548
- package options, 19, 145, *see also* specific packages, 253, 262, 307–308, 310, 316–318, 427, 543
 - are passed down from document classes, 254
 - preventing loading of, 309
- packages, 34, 253
 - accents, 173
 - `alltt`, 127
 - `amssys`, 310, 311
 - `amscd`, 244–247, 310
 - `amsmath`, 310–311
 - `amsgen`, 310, 311
 - `amsmidx`, 311
 - `amsmath`, 34, 236, 307–309, 311, 447
 - `amsopn`, 310, 311
 - `amssymb`, 178, 303, 309, 310, 415
 - `amstext`, 310, 311

- amsthm, 34, 311
 - amsxtra, 173, 309, 310
 - apalike, 481
 - automatic loading of, 344
 - babel, 145–146
 - backref, 316, 317
 - booktabs, 121
 - calc, 447, 450
 - commands in, 427
 - dcolumn, 116
 - distribution, AMS, 279, 309–311, 321, 421, 461, 480, 481
 - documentation for, 94, 145, 354, 421
 - enumerate, 103, 424, 428
 - eucal, 310, 311, 427, 543
 - options, 310, 433, 543
 - eufrak, 310
 - exscale, 310
 - fancyhdr, 278
 - fontenc, 145, 146
 - geometry, 43, 277
 - glossaries, 514
 - graphicx, 27, 192, 254, 265–266, 309
 - hhline, 120
 - hyperref, 316–319
 - inputenc, 145
 - interdependencies of, 279
 - L^AT_EX, 34, 56, 254
 - latexsym, 178, 253, 303, 309
 - layout, 275
 - loading with `\usepackage`, 56, 193, 253
 - multiple, 254
 - lucidabr, 145
 - makeglos, 514
 - makeidx, 500
 - mathtools, 209
 - multicol, 81
 - newlattice, 254, 427, 428
 - options, *see* package options, *see* package options
 - options
 - setspace, 78
 - showidx, 503
 - showkeys, 262
 - source files, 254, 481
 - stix, 193
 - TikZ, 357
 - times, 143, 144
 - upref, 309, 310
 - verbatim, 73, 74, 127, 428
 - xcolor, 321, 344
 - xspace, 408–409
 - xy-pic, 247
 - page (counter), 443
 - pagebackref (opt. of hyperref pack.), 316, 317
 - `\pagebreak` (page breaking com.), 80, 527, 528
 - optional arguments of, 80
 - `\pagename` (redef. name), 417
 - `\pagenumbering` (page-number style com.), 445
 - `\pageref` (cross-ref. com.), 20, 23, 259, 264, 274, 318, 527
 - `\pageref*` (hyperref cross-ref. com.), 318
 - pages
 - breaking, 80–81, 268
 - in `align` environment, 247
 - in cases subsidiary environment, 247
 - in displayed math environments, 247–248
 - in gather environment, 247
 - in multiline math environments, 248
 - preventing, 247
 - components of, 275
 - headers, *see* running heads
 - layout of, 275–278
 - for `amsart` document class, 276
 - numbering, style of, 445
 - odd, starting chapters on, 518
 - referencing, 264
 - size changing with `\enlargethispage`, 80, 527, 530
 - styles of, 275–278
 - title
 - document class options for, 253, 308, 309
 - of articles, 253, 255
 - of books, 517, 518
 - of presentations, 28
- pages (BIB_TE_X database field), 469
 - page ranges in, 471
- `\pagestyle` (set page style), 275
- pagination, 527–528
- paper size, document class options for, 307
- `\par` (paragraph breaking com.), 9, 50, 51, 60, 135, 281
- `\par` (paragraph-breaking com.), 98, 105, 142
 - in error messages, 79
- `\paragraph` (struct. com.), 256, 516
- paragraph (counter), 443
- paragraph boxes, 91–93
- paragraph text symbol, *see* `\P`
- `\paragraph*` (struct. com.), 257
- `\paragraphname` (hyperref redef. name), 319
- paragraphs, 50, 79–80
 - breaking into lines, 11, 75, 77
 - breaking lines in, 68
 - hanging indents, 79
 - horizontal alignment of, 104
 - indentation of first lines, 79, 83, 448
 - interline spacing of, 44
 - terminating
 - with blank lines, 9, 43, 50, 79, 98, 124

- with `\par` command, 50, 51, 60, 79, 105, 135, 142
 - vertical spacing between, 105, 448, 451, 457, 524
- `\parallel` (\parallel binary rel.), 533
- `\parbox` (box com.), 86, 90–92, 118, 449
 - in tabular environments, 92
- `parbox.tex` (sample file), 88
- `parentequation`, 445
- parentheses (`()`), 7, 48
 - as math delimiters, 16, 18, 43, 163, 183, 221, 539
 - in index entries, 504
 - suppressing the use of in tags, 206
- `\parindent` (length com.), 448, 459
- `\parsep` (length com.), 451, 452, 457, 458
- `\parskip` (length com.), 448, 451, 452, 459, 525
- `\part` (struct. com.), 257, 346, 516
- `part` (counter), 443
- `\partial` (∂ math sym.), 538
- `\partname` (redef. name), 417
 - and `hyperref` package, 319
- `\partopsep` (length com.), 451
- parts
 - in books, 516
 - in presentations, 349
- Patashnik, Oren, 465, 490, 561
- `\pause` (beamer com.), 31, 82, 327, 330, 333
- `pc` (point abs. unit), 447
- Windows computers, 5
- PDF (Portable Document Format), 529
 - files
 - bibliographic citations in, 317
 - bookmarks in, 317
 - external hyperlinks in, 318–319
 - for presentations, 28, 321–355
 - hyperlinks in, 315–319
 - suppressing hyperlinks in, 318
 - typeset, 6
- `PDFtoText`, 497
- percent (%)
 - as comment character, 8, 47, 71–73, 121, 303, 508
 - in \LaTeX databases, 72, 487
 - as text symbol, 7, 63, 67, 548
 - in e-mail addresses, 286
- periods (`.`), 7, 48, 52–53
 - and initials, 53
 - and italic correction, 137
 - and spacing rules, 52–53
 - end of sentences, 52
 - following capital letters, 53
 - in abbreviations, 52
 - in bibliographies, 53
 - in \LaTeX databases, 470
 - terminating sentences with, 50–53
- `\perp` (\perp binary rel.), 533
- personalized templates
 - for AMS document classes, 303–306
- `\phantom` (spacing com.), 82, 189, 191–192, 228, 450
- PHDTHESIS (bibl. entry type), 469, 477–478
- `\Phi` (Φ Greek char.), 162, 532
- `\phi` (ϕ Greek char.), 162, 532
- `\Pi` (Π Greek char.), 162, 532
- `\pi` (π Greek char.), 162, 532
- `pica` (pc abs. unit), 447
- `picture` (drawing env.), 265
- `pilcrow` (\P text sym.), 67, 548
- `\pitchfork` (\pitchfork binary rel.), 534
- placement
 - of commas in formulas, 153
 - of equation numbers, 307
 - of equations, 307
 - of figures, 267, 527, 529
 - of `\index` commands, 508, 528
 - of index in document, 500, 511
 - of `\label` commands, 528
 - of limits, 308
 - of roots, 160
 - of tables, 527, 529
 - of text boxes, 95
- `plain`
 - \LaTeX style, 465
 - page style, 275
 - theorem style, 109–111
- `plain.bst` (\LaTeX style file), 465
- platform independence
 - of graphics, 321
- plus (`+`), 15, 188, 191, 192, 220–222
 - as binary operation, 536
- plus and minus rule, 188, 191, 192
 - and subformulas, 220
 - in multiline environments, 217
- `\pm` (\pm math op.), 536
- `\pmatrix` (subsid. math env.), 17–18, 240
- `\pmb` (poor man's bold font com.), 202, 310
- `\pmod` (`(mod)` math op.), 16, 169
- `\pod` (`()` math op.), 16, 169
- poetry, typing, 124–125
- point (pt abs. unit), 12, 42, 76, 85, 87, 91, 92, 134, 138, 142, 447
- poor man's bold, 202, 310
- portability
 - of bibliographic databases, 470, 472
 - of prohibited characters, 49
- Portable Document Format, *see under* PDF
- Portable Graphics Format, 321
- position, of proclamation numbers, 111

- PostScript (PS)
 - fonts, 143
 - Computer Modern, 33, 143
- `\pounds`
 - math symbol (\pounds), 538
 - pound sign or sterling (\pounds), 549
- `\Pr` (Pr math op.), 168, 540
- preamble, 25–254
 - commands in, 19, 25, 28, 56, 71, 72, 78, 105, 107, 109, 111, 154, 200, 244, 253, 254, 303, 309, 316, 322–323, 406, 415, 423, 427, 446, 512, 529
 - custom commands in, 254, 406
 - customizing, in template files, 303
 - defining counters in, 444
 - proclamations defined in, 254
- `\prec` (\prec binary rel.), 533
- `\precapprox` (\preccurlyeq binary rel.), 534
- `\preccurlyeq` (\preccurlyeq binary rel.), 534
- `\preceq` (\preceq binary rel.), 533
- `\precnapprox` (\preccurlyeq neg. binary rel.), 535
- `\precneqq` (\preccurlyeq neg. binary rel.), 535
- `\precnsim` (\preccurlyeq neg. binary rel.), 535
- `\precsim` (\preccurlyeq binary rel.), 534
- preparation
 - final, of books, 525–530
 - of illustrations, 526
- presentations, 28–31, 321
 - `babybeamer1`, 327–332
 - `babybeamer2`, 335
 - `babybeamer3`, 336
 - `babybeamer4`, 337
 - `babybeamer5`, 340–342
 - `babybeamer6`, 343
 - `babybeamer7`, 346
 - `babybeamer8`, 354, 355
 - `beamerstructure1`, 347
 - `beamerstructure2`, 349
 - body of, 323
 - columns in, 342–343
 - commands for, 29, 344, 346, 352
 - optional arguments of, 347
 - cross-referencing within, 324
 - flexibility in, 353
 - frames in, 353
 - hyperlinks in, 338–342, 353
 - in color, 343–345
 - lists in, 335–336
 - navigation symbols, 349
 - notes in, 350, 351
 - overlays in, 327–338
 - commands for, 31, 327, 331, 334
 - examples of, 332–333
 - layering, 330, 332
 - specifications, 333, 336
 - syntax of, 334–335
 - parts in, 349
 - planning, 353
 - preamble of, 28, 322–323
 - `quickbeamer1`, 325, 326
 - `quickbeamer2`, 327, 328
 - sectioning of, 324
 - sidebars, 324
 - sidebars in, 338, 346
 - slides, 28, 327
 - structure of, 346
 - tables of contents in, 347, 353
 - themes for, 28, 324, 327, 351, 352
 - color options, 352
 - font options, 352
 - theorems in, 353
 - top matter of, 28, 322–323, 327
 - with beamer, 28
 - with beamer document class, 321–355
- preventing
 - case changes in bibliographic entries, 471
 - hyphenation, 69–70, 87
 - indentation of paragraphs, 79
 - line breaks, 78, 86
 - page breaks, 77, 80, 247
 - removal
 - of horizontal space, 83
 - of vertical space, 85
 - spaces following environments, 424
- `\prime` (\prime math sym.), 538
- prime (\prime), 15, 156, 198, 538
- primitive commands, 418
- printers
 - drivers for, 36
- `\Printindex` (index com. for multiple indexes), 513
- `\printindex` (index com.), 500, 511
- `proc-l` (doc. class), 417
- PROCEEDINGS (bibl. entry type), 469, 476–477, 480
- proceedings, conference, in BIBTEX
 - database files, 468
- processed index files, *see* ind
- processing
 - bibliographies, 481–489
 - indexes, 509–511
- proclamations, 42, 97, 105–111, 451
 - commands for
 - `\newtheorem` (procl. com.), 105, 107, 109, 111, 303, 446
 - `\newtheorem*` (procl. com.), 109, 423
 - `\newtheoremstyle`, 111
 - `\theoremstyle` (procl. com.), 109–111
 - `\theoremstyle*` (procl. com.), 109–111
 - counters for, 443

- defining, 105, 107, 108, 111, 303, 423, 446
 - in preamble, 254
- environments for
 - `theorem`, 37, 105, 108, 338, 425
- invoking, 37, 105, 108, 338, 425
- lines following, 43, 98
- lists in, 106
- names of, 108
- numbering of
 - consecutively, 107–108
 - within sections, 107
- position of numbers, 111
- styles of, 109–111
 - definition, 109–110
 - plain, 109–110
 - remark, 109–110
 - unnumbered, 109, 110, 423
- `\prod` (\prod large math op.), 171, 542
- products, 18, 170
- Professional Coder, 555
- Programming (GPT SP category), 555
- programs, typing, 121–123, 125–127
- prohibited keys, 7, 49
- `\projlim` (\projlim math op.), 168, 540
- prompt engineering, 553
- prompts
 - * (interactive), 94, 418–419
- `proof` (text env.), 97, 111–113, 311, 323
 - changing q.e.d. symbol, 415
 - lines following, 43, 98
 - lists in, 111
 - optional arguments of, 112
 - placement of q.e.d. symbol, 112
- `\proofname` (redef. name), 417
- proofs, 111–113
- proportional fonts, 133, 134
- propositions, *see* proclamations
- `\propto` (\propto binary rel.), 533
- `\protect` (protect fragile commands), 60, 256, 257, 516, 521
- protecting
 - capitalization in bibliographic entries, 471
 - fragile commands, 60, 256, 257, 516, 521
- `\providecommand` (custom com.), 413, 415, 419
- `\providecommand*` (custom com.), 413
- providing commands, 415
- `psfonts.ins` (PSNFSS distr. file), 144
- `\Psi` (Ψ Greek char.), 162, 532
- `\psi` (ψ Greek char.), 162, 532
- PSNFSS (PostScript New Font Selection Scheme)
 - distribution, 143
- pt (point abs. unit), 12, 42, 76, 85, 87, 91, 92, 134, 138, 142, 447
- `publisher` (BIBTEX database field), 468, 469
- publishers, preparing manuscripts for, 515
- punctuation
 - in BIBTEX databases, 467–472, 488–489
 - marks, 7, 11, 43, 48, 61–62, 67, 408, 545, 548
- q.e.d. symbol, 112, 415
- `\qedhere` (proof com.), 43, 113
- `\qedsymbol` (\square math sym.), 111, 415
 - suppressing, 112
- `\qqquad` (spacing com.), 81, 85, 190, 228, 544, 550
- `\quad` (spacing com.), 19, 81, 85, 152, 190, 229, 544, 550
- question marks (?), 7, 48
 - Spanish ($\text{\textcircled{?}}$), 67, 545, 548
 - terminating sentences with, 50–53
- `quickbeamer.tex` (sample file), 322–323
- `quickbeamer1.tex` (sample file), 323
- `quickbeamer2.pdf` (sample file), 327
- `quickbeamer2.tex` (sample file), 324, 327
- quotation (disp. text env.), 124, 451
- quotation marks, 7, 49, 61, 67
 - double quote key, 7, 48, 49, 61
 - in BIBTEX database fields, 489
 - in BIBTEX entries, 480
 - in `\index` commands, 507
 - European, 549
 - single quote key, 7, 15, 48, 61
 - typing, 9
- quotations, typing, 123–124
- `quote` (disp. text env.), 43, 123, 451
- `\quotedblbase` (Eur. quot. mark), 549
- quoting, special characters in `\index` commands, 507
- `\r` ($^{\circ}$ ring text accent), 66, 546
- `\raggedleft` (align. command dec.), 105
- `\raggedright` (align. command dec.), 105
- Rahtz, Sebastian, 265, 315, 559
- `\raisebox` (box com.), 95, 193
- ranges, numeric, 11, 62, 67, 548
 - in BIBTEX databases, 468, 471
 - in index entries, 504
- `\rangle` (\rangle math delimiter), 163, 164, 539
- `\rbrace` ($\}$ math delimiter), 43, 163, 539
- `\rbrack` ($\}$ math delimiter), 163, 539
- `\rceil` (\rceil math delimiter), 163, 539
- `\Re` (\Re math sym.), 538
- read-only, making templates, 306
- readability
 - of documents, 138, 353
 - of source files, 14, 15, 51, 115, 152, 178, 403–407, 421, 459
- Reader, Adobe Acrobat, *see under* Adobe Reader

- real numbers, in length commands, 447
- rectangle
 - draw (`\draw rectangle`), 369
- redefinable names, 145, 318, 319, 403, 416–417
- redefining commands, 414–415
- `\ref` (cross-ref. com.), 20, 23, 42, 72, 103, 153, 219, 259, 310, 318, 425
- `\ref*` (`\hyperref` cross-ref. com.), 318
- references
 - absolute, 21, 99, 206, 263
 - bibliographic, 53, 268–274, 324, 415, 451, 481
 - citing, 20, 273
 - citing with `BIBTEX`, 482
 - including without citing, 482
 - defining, 20, 23, 39, 42, 153, 219, 235, 256, 259, 324
 - symbolic, 72
 - symbolic, showing in margins, 262
 - to equations, 20, 23, 42, 153, 154, 207, 219, 223, 259, 260, 272
 - to pages, 20, 23, 259, 264, 318, 527
 - to sections, 20, 23, 103, 153, 219, 259, 310, 318, 425
- referencing, *see* cross-referencing *or* symbolic referencing, *see* cross-referencing *or* symbolic referencing
- `\refname` (redef. name), 417
- `\refstepcounter` (cross-ref. counter incr. com.), 446
- registered trademark (`\R` text sym.), 67, 548
- registers, 419
- relations, binary, *see* binary relations
- relative
 - numbering of equations, 205, 206, 263
 - spaces, 85
 - units, 85, 87, 92, 447
- `\relax` (do nothing com.), 426
- release dates, of `LATEX`, 72, 254
- remark (procl. style), 109–111
- `\renewcommand` (custom com.), 41, 74, 75, 78, 103, 112, 119, 274, 318, 410, 413–417, 419, 423, 424, 445–447
- `\renewcommand*` (custom com.), 413
- `\renewenvironment` (custom com.), 424, 426
- `\renewenvironment*` (custom com.), 426
- renumbering, automatic, 20, 36
- report (doc. class), 257, 312, 417
- reports, technical, in `BIBTEX` database files, 478–479
- reqno
 - `amsmath` package option, 311
 - document class option, 19, 253, 307
- required
 - arguments, 92, 93, 175, 197, 280, 281
 - multiple, 14, 56, 117, 194, 196, 197, 277, 420, 448–450
 - of commands, 14, 55, 117, 194, 196, 197, 261, 277, 287, 448, 520, 546
 - of environments, 56, 115, 116, 228, 230, 241, 242, 271–273, 425–427, 522
 - use braces (`{ }`), 6, 11, 14, 55, 56
- bibliographic fields, 470, 472–480
- `\RequirePackage` (pack. com.), 193, 427
- research support (`\thanks` top matter com.), 74, 287, 288, 305, 312
- resetting counters, 444, 446
- Return key, 7, 9, 49, 151
- `\rfloor` (`\lfloor` math delimiter), 163, 539
- `\rhd` (`\triangleright` math op.), 536
- `\rho` (ρ Greek char.), 162, 532
- `\right` (math delim.), 43, 164–166, 183, 240
 - blank, 164, 181
 - must be balanced, 164, 221
- right double quote (`"`)
 - text symbol, 61, 67, 548
 - typing, 9
- right justification of text, 10, 54
- right single quote (`'`)
 - key, 7, 9, 48, 61
 - text symbol, 61, 549
- `\right)` (`\)` math delimiter), 16, 18, 43, 163, 164, 183, 221, 240
- `\right.` (blank math delim.), 164
- `\right>` (`\rangle` math delimiter), 164
- `\right]` (`\]` math delimiter), 16, 163, 180
- `\Rightarrow` (\Rightarrow math arrow), 537
- `\rightarrow` (math arrow), 537
- `\rightarrowtail` (\rightarrowtail math arrow), 537
- `\rightharpoondown` (\rightarrowtail math arrow), 537
- `\rightharpoonup` (\rightarrowtail math arrow), 537
- `\rightleftarrows` (\rightleftarrows math arrow), 537
- `\rightleftharpoons` (\rightleftharpoons math arrow), 537
- `\rightmargin` (length com.), 451, 452, 455
- `\rightrightarrows` (\rightrightarrows math arrow), 537
- `\rightsquigarrow` (\rightsquigarrow math arrow), 537
- `\rightthreetimes` (\times math op.), 536
- ring (`\ring` text accent), 66, 546
- ring A (\AA), 67, 545
- ring a (\aa), 67, 545
- `\risingdotseq` ($\dot{=}$ binary rel.), 534
- `\rm` (obs. `LATEX 2.09` font com.), 140
- `\rmfamily` (font shape com. dec.), 104, 135, 546
- `rmfamily` (font shape env.), 104
- `\Roman` (uppercase roman-numeral counter-style com.), 445
- roman

- document font family, 134, 135, 143
- font shape, 20, 104, 133, 135, 310, 546
 - math, 199, 416
- `\roman` (lowercase roman-numeral counter-style com.), 103, 445
- roots, 18, 159–160
 - n -th, 18, 159
 - placement of, 160
 - square, 18, 159
 - using struts with, 203
- Rose, Kristoffer H., 247
- row separator (`\`), 17, 21, 23, 25, 77, 104, 115, 121, 123, 125, 172, 215, 216, 219, 220, 231, 235, 247, 281–285, 306
- `\Rrightarrow` (\Rightarrow math arrow), 537
- `\Rsh` (\r math arrow), 537
- `\rtimes` (\rtimes math op.), 536
- rubber lengths, 419, 450, 451
- `\myrule` (box com.), 94, 95, 120
- rules (lines), filling lines with, *see also* lines (rules), 85, 94
- running
 - BIBTEX, 481–489
 - MakeIndex*, 509–511
- running heads, 256, 275–278
 - author’s name in, 283
 - fragile commands in, 60
 - title in, 282, 306
- `\S` (§)
 - math symbol, 538
 - section text symbol, 67, 548
- sample files
 - 6colorpages, xxvi, 327, 329, 330, 345, 399
 - amsart.tpl, 303
 - amsproc template, 306
 - art1.tex, 25, 37, 38, 280, 322, 324
 - art1i1l1.tex, 27
 - art1ind.tex, 502, 510, 511
 - art2-ref.tex, 316, 319
 - art2.tex, 105, 109, 111, 234, 274, 280, 281, 293–299, 412, 436, 481
 - source file, 293–299
 - typeset, 293–302
 - art2biblatex.tex, 492
 - art2bibtex.tex, 465, 481, 483, 484, 487
 - art2ccom.tex, 436
 - art3.tex, 492, 495
 - babybeamer1.pdf, 327–332
 - babybeamer2.tex, 330, 334, 336
 - babybeamer3.tex, 336
 - babybeamer4.tex, 337, 339
 - babybeamer4block.tex, 338
 - babybeamer5.tex, 339–342
 - babybeamer6.tex, 342
 - babybeamer7.tex, 344
 - babybeamer8.tex, 354
 - beamerstructure.tex, 347, 349
 - bibtexcross.tex, 496
 - cleardoublepage.sty, 81
 - covers.pdf, 27
 - firstpresentation.tex, 28
 - fonttbl.tex, 64, 114
 - formulanote.tex, 12–14, 40–41
 - german.tex, 145
 - ggamsart.tpl, 306
 - inbibl.tpl, 271
 - myams.tpl, 303–306
 - newlattice.sty, 421, 424, 427–436
 - obtaining, 5
 - of presentations, 322–327
 - parbox.tex, 88
 - quickbeamer.tex, 322–323
 - quickbeamer1.tex, 323
 - quickbeamer2.pdf, 327
 - quickbeamer2.tex, 324, 327
 - sample.cls (doc. class), 8, 9
 - SymbolTables.pdf, 11, 14, 531–540, 542–550
 - textnote1.tex, 8–10
 - textnote1bad.tex, 11
 - textnote2.tex, 10–11
 - topmat.tpl, 289
- sample.cls, 9
- samples
 - of bibliographies, 268–271, 467, 481
 - of command (style) files, 421, 424, 427–436
 - of indexes, 500–502
- samples (folder), 5, 8, 10, 12, 13, 25, 28, 64, 81, 114, 145, 194, 234, 271, 289, 293, 303, 306, 316, 322, 323, 327, 335, 337–339, 349, 353, 354, 427, 436, 481, 500, 518
 - creating, 5
- sans serif
 - document font family, 134, 139, 143
 - font shape, 104, 134, 135, 199, 543, 546
 - Helvetica, 135, 143, 144
- `\sb` (math subscript com.), 156, 157
- `\sc` (obs. L^AT_EX 2.09 font com.), 140
- scaling graphics
 - in figures, 265
- Schöpf, Rainer, 561
- scharfes s (ß, SS), 67, 545
- Schleyer, Antoinette Tingley, 221, 561

- school (BIBTeX database field), 469
- scope
 - delimited by braces, 58–59
 - of `\allowdisplaybreaks` commands, 247
 - of command declarations, 105, 407–408, 423
 - of commands, 58–59, 120, 135
 - global and local, 60
 - setting, with environments, 76, 217
- `\scope`, TikZ environment, 372
- `\scriptscriptstyle` (math font size com.), 203, 207
- `\scriptsize` (font size com.), 138, 139, 547
- `\scriptstyle` (math font size com.), 203, 207
- `\scshape` (font shape com. dec.), 104, 135, 136, 546
- `scshape` (font shape env.), 104
- `\searrow` (\searrow math arrow), 537
- `\sec` (sec math op.), 168, 540
- `secnumdepth` (counter), 447
- `\section` (struct. com.), 255–259, 327, 346, 517
 - optional argument of, 257
- `section` (counter), 443
- section (§ text sym.), 67, 548
- `\section*` (struct. com.), 257, 517
- sectioning
 - commands
 - optional arguments of, 257
 - provided by `amsart` doc. class, 258
 - provided by `article` doc. class, 258
 - provided by book doc. classes, 516–517
 - syntax of, 257
 - of articles, 255–259
 - of books, 516–517, 523
 - of documents, 9, 25–259, 268, 274, 292, 451, 481, 499, 509, 516–517
 - of presentations, 324
- `\sectionname` (redef. name)
 - and `hyperref` package, 319
- sections, numbering of, 256
 - in books, 516
 - of equations within, 19, 154, 254
 - of proclamations within, 107
- `\see` (indexing cross-ref. com.), 507
- `\seename` (redef. name), 417
- `\selectfont` (font selection com.), 142
- selecting
 - files to include, 266, 444
 - fonts, 133–146
- semicolon (:), 7, 48
- sentences
 - terminating, 9, 50–53
 - with periods, 287
- `series` (BIBTeX database field), 469
- series commands, for fonts, *see under* font commands *and* font command declarations
- serif (font shape), *see also under* roman, 134
- Seroul, Raymond, 561
- `\setbeamer` (beamer com.), 350
- `\setbeamercolor` (beamer com.), 345
- `\setcounter` (counter-setting com.), 60, 238, 443, 446
- `\setlength` (length setting com.), 42, 79, 94, 448
- `setlength` environment, 76, 217
- `\setminus` (\setminus math op.), 536
- `setspace` (pack.), 78
- `\settodepth` (measurement com.), 450
- `\settoheight` (measurement com.), 450
- `\settowidth` (measurement com.), 450
- `\sffamily` (font shape com. dec.), 104, 135, 139, 546
- `sffamily` (font shape env.), 104
- shape commands, for fonts, *see under* font commands *and* font command declarations
- `\sharp` (\sharp math sym.), 538
- short
 - arguments, of custom commands, 413–414
 - arguments, of custom environments, 426
 - commands, 60, 135, 281, 292
- shorthand, *see under* custom commands
- `\shortmid` (\mid binary rel.), 534
- `\shortparallel` (\parallel binary rel.), 534
- `\shoveleft` (align. com.), 218
- `\shoveright` (align. com.), 218
- `\show` (command-examination com.), 418, 419
- `\showhyphens` (hyph.-disp. com.), 70
- `showidx` (pack.), 503
- `showkeys` (pack.), 262
 - options, 262
- `\showthe` (value-examination com.), 94, 418, 419
- shrinkable lengths, 80, 81, 419, 450, 451, 457
- side-setting math symbols, 197–198
- sidebars, in `beamer` presentations, 324, 338
- `\sideset` (math com.), 197–198
- `\Sigma` (Σ Greek char.), 162, 532
- `\sigma` (σ Greek char.), 162, 532

- `\sim` (\sim binary rel.), 533
- `\simeq` (\simeq binary rel.), 533
- simple alignment, of formulas, 21–24
- `\sin` (sin math op.), 18, 167, 168, 200, 540
- single
 - guillemet, 549
 - quote, 9
 - keys, 7, 15, 48, 61
 - text symbols, 549
- single quotation, 549
- single-lined boxes, *see under* boxes
- `\sinh` (sinh math op.), 168, 540
- size
 - of fonts, 42, 104, 134, 138–547
 - commands for, *see under* font commands
 - document class options for, 306
 - in math, 203
 - of hyperlinks in PDF files, 318
 - of math delimiters, 165–166, 183
 - of paper, document class options for, 307
- `\sl` (obs. L^AT_EX 2.09 font com.), 140
- slanted (font shape), 104, 133, 135, 136, 139, 140, 153, 154, 546
 - and italic correction, 138
 - context dependence of, 136
- slashed L's and O's (l, L, ϕ , \emptyset), 67, 545
- slides, *see under* presentations, *see under* presentations
- `slides` (obs. doc. class), 312
- `\slshape` (font shape com. dec.), 104, 135, 136, 138–140, 546
- `slshape` (font shape env.), 104
- slugs, 75, 76, 253, 309
- `\SMALL` (font size com.), 138, 139, 547
- `\Small` (font size com.), 138, 139, 547
- `\small` (font size com.), 138, 139, 547
- small caps (font shape), 104, 133, 135, 136, 546
 - for abbreviations and acronyms, 136, 524
- `\smaller` (font size com.), 139
- `\smallfrown` (\frown binary rel.), 534
- `\smallint` (\int math sym.), 538
- `smallmatrix` (subsid. math env.), 240
- `\smallsetminus` (\setminus math op.), 536
- `\smallskip` (spacing com.), 42, 84
- `\smallsmile` (\smile binary rel.), 534
- `\smash` (spacing com.), 96, 204
 - optional argument of, 205
- `\smile` (\smile binary rel.), 533
- solid boxes, 94, *see also* struts
- solution (beamer distr. directory), 353
- sort keys, 505–506
 - case sensitivity of, 512
 - sorting, of index entries, 505–506
- source files, 5–7, 13, 25, 34, 40, 50, 68, 74, 125, 459, 511, 519
 - errors in, 40–41
 - putting on the Internet, 319
 - readability of, 14, 51, 152, 178, 403–407, 421, 459
 - structure of, *see also* document
 - typesetting, 5, 11, 20, 36, 37, 40, 49, 60
 - white space in, 14
- sources, for fonts, 143
- `\sp` (math superscript com.), 156, 157, 182
- space bar, 7, 9, 49, 321, 330
- spaces
 - at the beginning of a line, 51, 83
 - at the end of a line, 50, 51, 71
 - blue, *see also* tie, unbreakable spaces, nonbreakable spaces, 62
 - expanding, 85–86
 - filling lines with, 85
 - in arguments of commands, 72
 - `\bibitem`, 273
 - `\cite`, 273
 - `\index`, 511–512
 - in command definitions, 407
 - in command names, 55
 - in delimited commands, 420–421
 - in math environments, 151
 - in `tabular` environments, 115
 - in `\verb` commands, 128
 - in `verbatim` environments, 126
 - interword (\backslash), 9, 52, 57, 81, 190, 550
 - multiple, act as one, 14, 50, 57
 - separating words with, 9, 50, 424, 471
 - suppressing, 424, 508
 - terminating commands with, 14, 55, 152
 - typed for readability, 15, 51, 115, 407
 - unbreakable (\sim tie), 9, 49, 52, 62, 78, 550
 - absorb spaces, 63
 - in Bib_TE_X databases, 471
 - with cross-references, 262
 - visible ($_$), 9, 49
 - and `\verb*` command, 128
- spacing
 - between dots with `\hdotsfor` commands, 238
 - between math symbols, 187–192
 - commands, *see* spacing commands
 - determined by document classes, 81
 - horizontal

- in math, 14, 152–153, 187–192, 238, 241
 - in text, 9, 42, 81–83, 95, 106, 458, 544, 550
 - interword, 9, 50–53, 57, 62, 81, 424, 471, 550
 - preventing removal of, 83
- in text, 81–86
- intercolumn, 211, 213, 224
 - in aligned math environments, 228, 233
 - in **tabular** environments, 116
- interline, 44, 166, 247, 410, 525
 - adjusting, 77–142
 - double, 78
- rules
 - and delimited commands, 420–421
 - for commands, 55, 152
 - in arguments of `\index` commands, 511–512
 - in math, 14, 152–153, 187–192, 203–205
 - in text, 9, 14, 50–53, 81–86
 - in `\text` commands, 152
- vertical
 - adding to table of contents, 520
 - adjusting, 119, 524–525
 - adjusting with the **setspace** package, 78
 - in boxes, 96
 - in indexes, 274, 499
 - in math, 203–205
 - in text, 42, 83–85, 204, 419, 520, 525, 529
 - preventing removal of, 85
 - using the **setspace** package, 78
- spacing commands
 - avoiding direct use of, 524–525
 - horizontal, *see* horizontal spacing
 - vertical, *see* vertical spacing
- `\spadesuit` (\spadesuit math sym.), 538
- `\spbreve` (\sim math accent), 172, 543
- `\spcheck` (\surd math accent), 172, 182, 543
- `\spdddot` (\cdots math accent), 172, 543
- `\spddot` ($\ddot{}$ math accent), 172, 543
- `\spdot` ($\dot{}$ math accent), 172, 543
- special
 - braces, 13, 58, 59, 150, 151, 154
 - balancing, 59
 - characters, 11, 49, 63, 64, 545–549
 - in `\index` commands, 507
 - keys, 7, 12, 49, 63
- spelling checkers, 37
- `\sphen` (\sim math accent), 172, 543
- `\sphericalangle` (\sphericalangle math sym.), 538
- `split` (subsid. math align. env.), 212, 214, 234–236, 307
 - and `\allowdisplaybreaks` commands, 248
 - numbering of equations within, 234–235
 - rules for, 235
- splitting
 - formulas across lines, 216–218
- `\sptilde` (\sim math accent), 172, 543
- `\sqcap` (\sqcap math op.), 536
- `\sqcup` (\sqcup math op.), 536
- `\sqrt` (\sqrt{x} math com.), 14, 18, 159, 191
 - optional argument of, 56
- `\sqsubset` (\sqsubset binary rel.), 533
- `\sqsubseteq` (\sqsubseteq binary rel.), 533
- `\sqsupset` (\sqsupset binary rel.), 533
- `\sqsupseteq` (\sqsupseteq binary rel.), 533
- `\square` (\square math sym.), 538
- square roots, 18, 159
- `\SS` (SS), 67, 545
- `\ss` (ß), 67, 545
- stacking math symbols, 44, 194–197
- `\stackrel` (symbol-building com.), 44
- `\star` (\star math op.), 536
- start of chapters, document class options
 - for, 518
- `\stepcounter` (counter-incr. com.), 446
- sterling (\pounds text sym.), 549
- STIX
 - fonts, 193
 - math symbols, 187–192
- stix** (pack.), 193
- STIX math symbols, 194
- stretchable
 - horizontal braces, 173–174
 - lengths, 81, 419, 450, 451
 - lines, 175
 - math arrows, 175, 245–247
 - math delimiters, 43, 164–165
 - math symbols, 173–175
 - vertical spacing, in displayed text environments, 450
- STRING (bibl. entry type), 480
- structural commands
 - `\appendix`, 259, 517
 - `\backmatter`, 517
 - `\chapter`, 257, 259, 516, 517
 - `\frontmatter`, 517
 - hierarchy of, 256, 516
 - in presentations, 346
 - `\mainmatter`, 517
 - optional arguments of, 257
 - `\paragraph`, 256, 516
 - `\paragraph*`, 257

- `\part`, 257, 516
 - provided by `amsart` doc. class, 258
 - provided by `article` doc. class, 258
 - provided by book doc. classes, 516–517
- `\section`, 255–259, 517
- `\section*`, 257, 517
- `\subparagraph`, 256, 516
- `\subsection`, 256, 516, 517
- `\subsection*`, 517
- `\subsubsection`, 256, 516
 - syntax of, 257
- structures, theorem-like, *see* proclamations
- struts, 95, 119, 203
- style (`sty`) files, 254, 406, 427–436
 - commands in, 427
 - terminating, 436
- style files, `BIBTeX`, *see* `bst`
- styles
 - bibliographical, *see also* `bst`, 273
 - for counters, 74, 103, 424, 445–446, 455
 - of pages, 275–278
 - of proclamations, 109–111
 - of tables, 120–121
 - text environments, 104
- `subarray` (subsid. math env.), 159, 172
- subentries, in indexes, 274, 499, 503, 504, 506
- subequations (disp. math env.), 206, 207, 223, 445
- subformulas
 - indentation of, 221
 - rules for, 219–221, 227–228
 - shorthand commands for, 404–406, 412–413
- `\subitem` (index com.), 274, 499
- `\subjclass` (top matter com.)
 - optional arguments of, 287
 - rules for using, 287
- `\subjclassname` (redef. name), 417
- subject classifications, AMS, 287
- `\subparagraph` (struct. com.), 256, 516
- `subparagraph` (counter), 443
- subscripts, math, 15, 16, 156–157, 159, 180
 - as limits, 16, 159, 167, 169–170, 180, 181, 307, 416
 - as multiline limits, 170–172
 - double, 203
 - font size of, 203
 - with horizontal braces, 174
- `\subsection` (struct. com.), 256, 346, 516, 517
- `subsection` (counter), 443
- `\subsection*` (struct. com.), 517
- `\subsectionname` (`hyperref` redef. name), 319
- `\Subset` (\subseteq binary rel.), 534
- `\subset` (\subset binary rel.), 533
- `\subseteq` (\subseteq binary rel.), 533
- `\subseteqq` (\subseteq binary rel.), 534
- `\subsetneq` (\subsetneq neg. binary rel.), 535
- `\subsetneqq` (\subsetneqq neg. binary rel.), 535
- subsidiary math environments
 - `aligned`, 214, 232–234, 248
 - `alignedat`, 232–234, 244, 248
 - and `\allowdisplaybreaks`
 - commands, 248
 - `array`, 212, 214, 233, 237, 241–244
 - `Bmatrix`, 240
 - `bmatrix`, 240
 - `cases`, 24–25, 212, 214, 243–244
 - `CD`, 244–247
 - for aligning formulas, 212, 213, 231–236, 248, 307
 - `gathered`, 214, 232–235, 248
 - `matrix`, 17–18, 212, 214, 236–243
 - numbering of, 234, 235
 - page breaks in, 247
 - `pmatrix`, 17–18, 240
 - `smallmatrix`, 240
 - `split`, 212, 214, 234–236, 248, 307
 - `subarray`, 159, 172
 - `Vmatrix`, 182, 240
 - `vmatrix`, 17, 18, 240
- `\substack` (math com.), 159, 170, 172
- substitution, of fonts, 139, 527
- subsubentries, in indexes, 274, 499, 504, 506
- `\subsubitem` (index com.), 274, 499
- `\subsubsection` (struct. com.), 256, 516
- `subsubsection` (counter), 443
- `\subsubsectionname` (redef. name), 318
- subtraction, 15, 49, 155
- `\succ` (\succ binary rel.), 533
- `\succapprox` (\gtrsim binary rel.), 534
- `\succcurlyeq` (\gtrsim binary rel.), 534
- `\succeq` (\succeq binary rel.), 533
- `\succnapprox` (\gtrsim binary rel.), 535
- `\succneqq` (\gtrsim binary rel.), 535
- `\succnsim` (\gtrsim binary rel.), 535
- `\succsim` (\gtrsim binary rel.), 534
- `\sum` (\sum large math op.), 18, 171, 202, 542
 - with primes, 197
- `sumlimits` (opt. of `amsmath` pack.), 308
- sums, 18, 169, 170
- `\sup` (sup math op.), 168, 540

- superscripts, 15, 16, 156–157, 159, 173, 180
 - as limits, 16, 159, 167, 169–170, 180
 - as multiline limits, 170–172
 - in math
 - as limits, 307
 - font size of, 203
 - with horizontal braces, 174
 - in text, 549
- `\suppressfloats` (float com.), 267
- suppressing
 - creation of auxiliary files, 529
 - hyperlinks in PDF files, 318
 - indents (`\noindent` com.), 79, 527
 - italic correction, 137–138
 - ligatures, 64, 179, 405
 - line breaks (`\nolinebreak` com.), 78
 - loading of the `amsmath` package, 309
 - numbering, 215, 219
 - of equations, 22, 23, 154, 205, 215, 217, 219, 222, 235, 277
 - of first page of documents, 277
 - of proclamations, 109, 110, 423
 - of structural units, 257, 517
 - `Overfull \hbox` warnings, 76
 - page breaks (`\nopagebreak` com.), 80
 - placement of floats, 267
 - q.e.d. symbol, 112
 - spaces, 424, 508
- `\Supset` (\supseteq binary rel.), 534
- `\supset` (\supset binary rel.), 533
- `\supseteq` (\supseteq binary rel.), 533
- `\supseteqq` (\supseteqq binary rel.), 534
- `\supsetneq` (\supsetneq binary rel.), 535
- `\supsetneqq` (\supsetneqq binary rel.), 535
- `\surd` (\surd math sym.), 538
- Sutor, Robert, 559
- Swanson, Ellen, 221, 561
- `\swapnumbers` (theorem-style com.), 111
- `\swarrow` (\swarrow math arrow), 537
- `\symbol` (symbol com.), 64
- symbol alphabets, math, 200–201
 - blackboard bold, 201
 - calligraphic, 200
 - Euler Fraktur, 201
 - Greek, 200
- symbolic references, showing in margins, 262
- symbolic referencing, *see also* cross-referencing, *see also* cross-referencing, 20–72
- symbols
 - as footnote indicators, 74
 - math, 14, 40, 49, 111, 173, 181, 245, 253, 303, 161–544
 - alphabets, 200–201
 - and delimiters, 16, 163, 164
 - bold, 201–202, 310
 - building new, 181, 194–198, 412
 - classification of, 188–189, 199
 - declaring types of, 198
 - end of proof, 94, 415
 - in text, 405
 - large, 231–233
 - negated, 197–198
 - shorthand commands for, 404
 - side-setting, 197–198
 - sizes of, 203
 - spacing of, 187–192
 - stacking, 44, 194–197
 - STIX, 187–194
 - stretchable, 173–175
 - suppressing, 111, 112
 - text, 63, 65, 286, 546, 548, 549
- `SymbolTables.pdf` (sample file), 11, 14, 531–540, 542–550
- systems of equations, 229–230
- `\t` (tie text accent), 66, 546
- T1 font encoding, 141
- T1 font encoding, 549
- T1 font encodings, 69
- tab
 - character, 49, 50, 54, 81
 - Tab key, 7, 9, 49
- Tab key, 7, 9, 49
- tabbing (display text env.), 121–123
- table (counter), 443
- table (float env.), 264–266
 - optional arguments of, 267, 529
 - placement of, 527, 529
- table* (float env.), 266
- `\tablename` (redef. name), 417
 - and `hyperref` package, 319
- `\tableofcontents` (front-matter com.), 324, 519
- tables, 113–121, 242, 266, 267
 - captions in, 27, 114, 266, 522
 - fragile commands in, 60
 - commands for, 27, 115, 117–120, 266, 522
 - cross-referencing of, 114
 - designing, 120–121
 - double-column, 266
 - float controls, 114
 - forcing typesetting of, 80
 - intersection of lines in, 120
 - lists of, *see under* lists
 - multiline entries in, 118
 - numbering of, 264

- placement of, 527, 529
 - style of, 120–121
 - typeset inline, 113
- tables of contents, 256, 317, 519–521
 - adding a line to, 519–520, 530
 - adding vertical spacing to, 520
 - auxiliary file (`toc`), 519, 529
 - commands, 519, 520
 - depth of, 447
 - fragile commands in, 60, 521
 - in frames, 324
 - in presentations, 353
 - typesetting, 519–521
- tabs, setting, 121
- tabular** (table env.), 64, 113–120, 242, 266
 - arguments of, 115
 - optional, 115
 - breaking lines in, 115
 - column-formatting in, 116
 - horizontal lines in, 115, 117
 - intercolumn space in, 116
 - `\parbox` in, 92
 - rules for, 115
 - vertical lines in, 115
 - width of columns in, 116
- `\tag`, 21, 205, 206, 215, 217, 219, 222, 234, 235, 263
- `\tag*`, 206
- tags (names for equations), 21, 205–207, 215, 217, 219, 222, 234, 235, 263
 - top-or-bottom placement of, 307
- `\tan` (tan math op.), 168, 540
- `\tanh` (tanh math op.), 168, 540
- Tantau, Till, 321, 354, 561
- `\tau` (τ Greek char.), 162, 532
- `\tbinom` (inline-style binomial com.), 157
- `tbtags` (doc. class opt.), 307
- technical reports, in BibTeX database files, 478–479
- TECHREPORT (bibl. entry type), 469, 478–479
- templates, 279
 - bibliographical, 268, 481
 - customizing
 - for AMS document classes, 303–306
 - `gamsart.tpl`, 306
 - `myams.tpl`, 303–306
 - personalized
 - making read-only, 306
- terminating
 - captions, 522
 - commands, 11, 14, 55–57, 66, 128, 152
 - environments, 54
 - fields in BibTeX database entries, 489
 - files, 436
 - lines, 72, 156, 407
 - paragraphs, 43, 50, 79
 - sentences, 9, 50–53, 287
- TeX, 33, 34
 - commands, 41, 419, 421
 - to avoid using, 525
 - commands in LaTeX, 525
 - extensibility of, 33
 - inputs folders, 144
 - programming in, 33
- \TeX (TeX com.), 66
- TeX users groups, 145
- text, 6
 - accents, 55, 66, 546
 - blocks, width of, 277
 - boxes, *see* text boxes
 - centering, *see* centering text, `center` (text align. env.), *and* `\centering`, *see* centering text, `center` (text align. env.), *and* `\centering`
 - custom commands for, 406
 - editors, 5, 6, 13, 34, 38, 49, 70, 133, 405
 - framing, 90
 - in math mode, 14, 19, 69, 78, 86, 87, 152, 160–161, 179, 200, 230–231, 246, 310
 - specifying fonts for, 161
 - in tables
 - multicolumn, 117–120
 - multiline, 118
 - mode, 6
 - spacing in, 81–86
 - style commands, *see* text style commands
 - symbols, *see* text symbols
 - typing, 47–96, 149
 - using math symbols in, 405
 - white space in, 81–86
- `\text` (box com.), 14, 19, 69, 70, 78, 160–161, 179, 209, 246, 310
 - spacing rules in, 152
- text boxes, 86–96
 - alignment of contents in, 87, 92
 - behave as single characters, 86
 - commands for
 - `\framebox`, 90
 - `\makebox`, 86–88, 90, 92, 94
 - `\mbox`, 87
 - `\parbox`, 86, 90–92, 118, 449
 - `\raisebox`, 95, 193
 - `\myrule`, 94, 95, 120
 - `\mbox`, 86
 - `\text`, 14, 19, 69, 70, 78, 152, 160–161, 179, 209, 246, 310

- framed, 90–91
- measurements of, 449
- measuring, *see* measurements
- single line, 86–90, 160
- vertical alignment of, 92, 96
- text environments, 97–129
 - and stretchable vertical space, 450
 - blank lines in, 43, 98
 - for abstracts, *see* **abstract**
 - for alignment, 54, 86, 104, 114, 231, 451, 459
 - for boxes, *see* *under* text boxes
 - for documents, *see* **document**
 - for indexes, *see* **theindex**
 - for letters, *see* **letter**
 - for lists, *see* list text environments, *see* list text environments
 - for styles, *see* *under* text style
 - for tables, *see* tables
 - rules for, 98
 - thebibliography**, 481
 - theindex**, 499, 509
- text mode, 48
- text style commands
 - avoiding direct use of, 524
 - \emph**, 6, 11, 54, 135, 136, 334, 546
 - \textbf**, 11, 60, 135, 138, 161, 310, 334, 546
 - \textit**, 135, 136, 334, 546
 - \textmd**, 135, 138, 546
 - \textnormal**, 135, 161, 546
 - \textrm**, 135, 310, 546
 - \textsc**, 135, 136, 546
 - \textsf**, 135, 546
 - \textsl**, 135, 136, 546
 - \texttt**, 11, 63, 135, 546
 - \textup**, 135, 136, 546
- text style environments
 - bfseries**, 104
 - em**, 104
 - itshape**, 104
 - rmfamily**, 104
 - scshape**, 104
 - sffamily**, 104
 - slshape**, 104
 - ttfamily**, 104
 - upshape**, 104
- text symbols, 64, 67
 - commands, 63, 65, 286, 546, 548, 549
 - typing, 61–65
- \textasciicircum** (^ circumflex), 67, 548
- \textasciitilde** (~ tilde), 549
- \textasteriskcentered** (* asterisk), 63, 67, 548
- \textbackslash** (\ backslash), 63, 67, 548
- \textbar** (| vertical bar), 63, 67, 548
- \textbf** (font weight com.), 11, 60, 135, 138, 161, 310, 334, 546
- \textbullet** (• bullet), 67, 548
- \textcircled** (@), 65, 67, 548
- \textcolor** (beamer com.), 344
- \textcompwordmark** (lig.-suppr. com.), 64, 405, 459
- \textemdash** (— em dash), 67, 548
- \textendash** (– em dash), 67, 548
- \textexclamdown** (¡ exclamation mark), 48
- \textgreater** (> greater than), 67, 548
- \textit** (font shape com.), 135, 136, 334, 546
- \textless** (< less than), 67, 548
- \textmd** (font weight com.), 135, 138, 546
- \textnormal** (font shape com.), 135, 161, 546
- textnote1.tex** (sample file), 8–10
- textnote1bad.tex** (sample file), 11
- textnote2.tex** (sample file), 10–11
- \TextOrMath**, 436
- \textquestiondown** (¿ question mark), 67, 548
- \textquotedbl** (Eur. quot. mark), 549
- \textquotedblleft** (“ left double quote), 67, 548
- \textquotedblright** (” right double quote), 67, 548
- \textquoteleft** (‘ left single quote), 549
- \textquoteright** (’)
 - Eur. quot. mark, 549
 - right single quote, 549
- \textregistered** (® registered trademark), 67, 548
- \textrm** (font shape com.), 135, 310, 546
- \textsc** (font shape com.), 135, 136, 546
- \textsf** (font shape com.), 135, 546
- \textsl** (font shape com.), 135, 136, 546
- \textstyle** (math font size com.), 203, 207
- \textsubscript**, 254
- \textsuperscript** (ˆ a), 549
- \texttrademark** (™ trademark), 549
- \texttt** (font shape com.), 11, 63, 135, 546
- \textup** (font shape com.), 135, 136, 546
- \textvisiblespace**, 9, 49, 50, 549
- \textwidth** (length com.), 277, 443, 448
- \tfrac** ($\frac{x}{y}$), 155, 203
- \TH** (Thorn Eur. char.), 549
- \th** (thorn Eur. char.), 549
- \thanks** (top matter com.), 74, 287, 288, 305, 312
- \the** (value expansion com.), 66–68, 72, 74, 94, 445

- `thebibliography` (bibliography env.), 53, 268–274, 451, 481, 489
 - argument of, 271, 273
- `\thechapter` (the value of counter `chapter`), 445
- `\thefootnote` (the value of counter `footnote`), 74, 75
- `theindex` (index env.), 274, 451, 499, 509
 - commands in, 274, 499
- `themes` (beamer distr. directory), 351
- themes, for beamer presentations, 324, 351–352
 - Berkeley, 324, 327, 328, 351
 - color options, 352
 - Warsaw, 29, 327, 328, 351
- `theorem` (proclamation env.), 37, 105, 108, 338, 425
 - logical design of, 37
 - visual design of, 37
- theorem-like structures, *see* proclamations
- `\theoremname` (`hyperref` redef. name), 319
- `\theoremstyle*` (procl. com.), 109–111
- `\theoremstyle` (procl. com.), 109–111
- `\theparentequation` (the value of counter `parentequation`), 446
- `\therefore` (\therefore binary rel.), 534
- `\thesection` (the value of counter `section`), 445
- theses, in `BIBTeX` database files, 477–478
- `\thesubsection` (the value of `subsection`), 445
- `\Theta` (Θ Greek char.), 162, 532
- `\theta` (θ Greek char.), 162, 532
- `\thickapprox` (\approx binary rel.), 534
- `\thicksim` (\sim binary rel.), 534
- `\thickspace` (spacing com.), 190, 544, 550
- `\thinspace` (spacing com.), 16, 53, 61, 190, 544, 550
- `\thispagestyle` (set page style), 277, 530
- thorn (Eur. char), 549
- Thorup, Kresten K., 447
- `tie` (\sim spacing com.), 9, 49, 52, 62, 78, 550
 - absorbs spaces, 63
 - in `BIBTeX` databases, 471
 - with cross-references, 262
- `tie` (\sim text accent), 66, 546
- `TikZ` package, 357–399
- `TikZ` command
 - `\clip`, 379
 - `\coordinate`, 378
 - `\draw`, 358–397
 - `\filldraw`, 365
 - `\fill`, 365
 - `\foreach`, 374
 - `\node`, 376
- `TikZ` environment
 - `\scope`, 372
- `TikZ` library
 - `arrows.meta`, 362
 - `calc`, 387
 - `datavisualization`, 394
- `TikZ` transformation, 371
 - `rotate`, 373
 - `scale`, 371
 - `shift`, 379
- `tilde` (\sim)
 - text accent, 66, 546
 - text symbol, 549
- `\tilde` (\tilde math accent), 17, 172, 543
- `\time` (time com.), 66
- time commands
 - `\day`, 66
 - `\month`, 66
 - `\time`, 66
 - `\today`, 10, 56–57, 66, 128, 145, 421
 - `\year`, 66
- Times, in `LATEX` documents, 143
- `\times` (\times math op.), 15, 155, 536
- `times` (pack.), 143, 144
- `times.sty` (PSNFSS distr. file), 144
- `\Tiny` (font size com.), 138, 139, 547
- `\tiny` (font size com.), 138, 139, 547
- Tisseau, Gérard, 561
- `\title` (top matter com.), 29, 86, 253, 306, 312, 327, 518
 - optional arguments of, 29, 281, 282, 290, 306
- `title` (`BIBTeX` database field), 468–470, 477
- title pages, 281
 - document class options for, 253, 308, 309
 - footnotes on, 74
 - of articles, 253, 255
 - of books, 517, 518
 - of presentations, 28
- `titlepage` (doc. class opt.), 253, 308, 518
- `\titlepage` (beamer com.), 324
- titles
 - in bibliographies
 - rules for typing, 470–472
 - in running heads, 282, 306
 - of articles, 281, 306
 - of structural commands, 256
 - fragile commands in, 60
- `\to` (\rightarrow math arrow), 537
- `toc` (table of contents files), 519–521, 529
 - commands in, 521
- `tocdepth` (counter), 447

- `\today` (time com.), 10, 56–57, 66, 128, 145, 421
- token, 557
- `\top` (\top math sym.), 538
- top matter, 26, 253, 255, 280
 - article information
 - of AMS documents, 281–283
 - commands, 29, 66, 74, 86, 277, 281–290, 292, 305, 306, 327, 518
 - blank lines in, 281
 - examples of, 289–292
 - with multiple authors, 288–289
- customizing templates, 303
- is document-class specific, 255
- of AMS documents, 281–292
 - AMS information, 287–288
 - author information, 283–287
 - errors with, 292
 - examples, 289–292
 - with multiple authors, 288
- of books, 255
- of presentations, 28, 322–323, 327
- top-or-bottom, placement of tags, 307
- `topmat.tpl` (sample file), 289
- `\topsep` (length com.), 451, 452
- `\totalheight` (length com.), 87, 88, 91, 92
- trademark text symbols ([™] ®), 67, 548, 549
- `trans` (beamer doc. class opt.), 353
- translation, of line ending characters, 49
- `\translator` (top matter com.), 282
- `\triangle` (\triangle math sym.), 538
- `\triangledown` (∇ math sym.), 538
- `\triangleleft` (\triangleleft math op.), 536
- `\trianglelefteq` (\trianglelefteq binary rel.), 534
- `\triangleq` (\triangleq binary rel.), 534
- `\triangleright` (\triangleright math op.), 536
- `\trianglerighteq` (\trianglerighteq binary rel.), 534
- `trivlist` (list text env.), 458
- `\tt` (obs. L^AT_EX 2.09 font com.), 140
- `\ttfamily` (font shape com. dec.), 104, 123, 135, 546
- `\ttfamily` (font shape env.), 104
- TUG (T_EX Users Group), 5
- two-letter font commands, *see under* font commands
- `twocolumn` (doc. class opt.), 71, 81, 265, 266, 308
- `\twocolumn` (double-column com.), 81
- `\twoheadleftarrow` (\twoheadleftarrow math arrow), 537
- `\twoheadrightarrow` (\twoheadrightarrow math arrow), 537
- `twoside` (doc. class opt.), 71, 253, 308, 518
- type (B^IB_TE_X database field), 469
- type foundries, 143
- types
 - of commands, 60–61
 - of math symbols, 188, 202
 - declaring, 198
- typeset
 - example articles
 - `art2.tex`, 293–302
 - files, 6
 - PDF format, 6, 265
- typesetting
 - indexes, 509–511
 - source files, 5, 11, 20, 36, 37, 40, 49, 60
 - tables of contents, 519–521
 - with L^AT_EX, 34, 35
- typewriter style
 - document font family, 134, 135
 - font shape, 11, 63, 65, 104, 123, 135, 143, 405, 546
 - fonts, 133
 - in math, 199, 543
- typing
 - bibliographic entries, 470–481
 - accents in, 472
 - names in, 470
 - rules for, 487–489
 - titles in, 470–471
 - books, 515–530
 - exercises, 522
 - footnotes, 74–75
 - math, 25, 209
 - poetry, 124–125
 - programs, 121–123, 125–127
 - quotations, 123–124
 - text, 47–96, 149
 - symbols, 61–65
- typographical
 - errors, 37
 - rules for
 - abbreviations and acronyms, 136
 - initials, 53
- `\u` breve text accent (\breve), 66, 546
- `\ulcorner` (\ulcorner math delimiter), 163, 539
- Umeki, Hideo, 277
- umlaut (\ddot text accent), 7, 11, 66, 145, 546
- unbreakable
 - hyphens, 69
 - spaces (\sim tie), 9, 49, 52, 62, 78, 550
 - absorb spaces, 63
 - in B^IB_TE_X databases, 471
 - with cross-references, 262
- `\underbrace` (math sym.), 174
 - with a subscript, 174
- underdot text accent (\dot), 66, 546
- `\underleftarrow` (\underleftarrow math arrow), 175

- `\underleftrightharrow` ($\mathrel{\mathop{\leftrightarrow}}^x$ math arrow), 175
- `\underline` (\underline{x} math line sym.), 175
- `\underrightharrow` ($\mathrel{\mathop{\rightarrow}}^x$ math arrow), 175
- `underscore` (`_`), 63
 - text symbol, 67, 548
 - in e-mail addresses, 286
- `underscore text accent` (`_`), 66, 546
- `\underset` (symbol-building com.), 181, 196
- units
 - absolute, *see under* absolute
 - dimensional, *see under* dimensional units
 - math, *see* μ
 - relative, *see under* relative
- UNIX
 - L^AT_EX implementations, 511
- `\unlhd` (\triangleleft math op.), 536
- unmarked footnotes, 287
- unnumbered
 - environments, 219
 - equations, 19, 22, 23, 154, 205, 214, 215, 217, 219, 222, 235, 324
 - items, 99
 - math environments, 215
 - proclamations, 109, 110, 423
 - structural units, 257, 517
- UNPUBLISHED (bibl. entry type), 479
- `\unrhd` (\triangleright math op.), 536
- `\Uparrow` (\Uparrow)
 - math arrow, 537
 - math delimiter, 163, 539
- `\uparrow` (\uparrow)
 - math arrow, 537
 - math delimiter, 163, 539
- updates
 - to L^AT_EX, 194
- `\Uparrow` (\Uparrow)
 - math arrow, 537
 - math delimiter, 163, 539
- `\updownarrow` (\updownarrow)
 - math arrow, 537
 - math delimiter, 163, 539
- `\upharpoonleft` (\lhd math arrow), 537
- `\upharpoonright` (\rhd math arrow), 537
- `\uplus` (\uplus math op.), 536
- uppercase counter styles
 - letters (`\alph`), 445
 - roman numerals (`\roman`), 445
- `upref` (pack.), 309, 310
- upright (font shape), 20, 104, 133–136, 153, 154, 424, 456, 546
 - context dependence of, 136
- `\uproot` (root-adjustment com.), 160
- `\upshape` (font shape com. dec.), 104, 135, 136, 424, 456, 546
- `upshape` (font shape env.), 104
- `\Upsilon` (Υ Greek char.), 162, 532
- `\upsilon` (υ Greek char.), 162, 532
- `\upuparrows` (\Uparrow math arrow), 537
- `\urcorner` (\urcorner math delimiter), 163, 539
- `\URL` (bibl. com.), 469
- URL (Uniform Resource Locator), 286, *see also* hyperlinks
- `\urladdr` (top matter com.), 286, 305, 319
- `\usecolortheme` (beamer com.), 352
- `\usecounter` (list counter com.), 455
- `\usefonttheme` (beamer com.), 352
- `\usepackage` (preamble com.), 25, 56, 253, 303, 316, 322, 406, 427, 500
 - can load multiple packages, 254
- user-defined commands, *see* custom commands
- users groups, 145
- `\settheme` (beamer com.), 28, 324, 351
- UTF-8 (Unicode Transformation Format) encoding, 141
- `\v` caron text accent (v^{c}), 66, 546
- `\value` (value of counter), 446
- values, of counters, 36, 74, 446
 - printing, 66–68, 94, 445
- van Oostrum, Piet, 278
- `\varDelta` (Δ Greek char.), 162, 532
- `\varepsilon` (ε Greek char.), 162, 532
- `\varGamma` (Γ Greek char.), 162, 532
- `\varinjlim` (\varinjlim math op.), 168, 540
- `\varkappa` (\varkappa Greek char.), 162, 532
- `\varLambda` (Λ Greek char.), 162, 532
- `\varliminf` (\varliminf math op.), 168, 540
- `\varlimsup` (\varlimsup math op.), 168
- `\varnothing` (\varnothing math sym.), 538
- `\varOmega` (Ω Greek char.), 162, 532
- `\varPhi` (Φ Greek char.), 162, 532
- `\varphi` (φ Greek char.), 162, 532
- `\varPi` (Π Greek char.), 162, 532
- `\varpi` (ϖ Greek char.), 162, 532
- `\varprojlim` (\varprojlim math op.), 168, 540
- `\varpropto` (\propto binary rel.), 534
- `\varPsi` (Ψ Greek char.), 162, 532
- `\varrho` (ϱ Greek char.), 162, 532
- `\varSigma` (Σ Greek char.), 162, 532
- `\varsigma` (ς Greek char.), 162, 532
- `\varsubsetneq` (\subsetneq neg. binary rel.), 535
- `\varsubsetneqq` (\subsetneqq neg. binary rel.), 535
- `\varsupsetneq` (\supsetneq binary rel.), 535
- `\varsupsetneqq` (\supsetneqq binary rel.), 535
- `\varTheta` (Θ Greek char.), 162, 532

- `\vartheta` (ϑ Greek char.), 162, 532
- `\vartriangle` (\triangle math op.), 536
- `\vartriangleleft` (\triangleleft math op.), 536
- `\vartriangleright` (\triangleright math op.), 536
- `\varUpsilon` (Υ Greek char.), 162, 532
- `\varXi` (Ξ Greek char.), 162, 532
- `\Vdash` (\Vdash binary rel.), 534
- `\vDash` (\models binary rel.), 534
- `\vdash` (\vdash binary rel.), 533
- `\vdots` (vertical ellipsis), 159, 240
- `\vec` (\vec{x} math accent), 17, 172, 418, 543
- `\vee` (\vee math op.), 536
- `\veebar` (\veebar math op.), 536
- `\verb` (inline verbatim com.), 127–129
 - in aligned math environments, 128
 - in argument of other commands, 128
 - spaces in, 128
- `\verb*` (inline verbatim com.), 128
 - delimiters with, 127–129
- `verbatim` (disp. text env.), 98, 125–127, 354
 - blank lines in, 126
 - characters following, 126
 - simulating with `\verb`, 129
- `verbatim` (pack.), 73, 74, 127, 428
- `verse` (disp. text env.), 124, 451
 - breaking lines in, 125
- `\vert` ($|$ math delimiter), 163, 539
- vertical
 - alignment, of boxes, 92, 96
 - bar ($|$ text symbol), 67, 548
 - ellipses, 159, 240
 - lines
 - in CD environments, 246
 - in `tabular` environments, 115
 - math arrows, 246
- vertical spacing
 - adding after `\`, 76
 - adding to table of contents, 520
 - adjusting
 - with `\arraystretch`, 119
 - with struts, 119
 - adjusting with the `setspace` package, 78
 - between marginal notes, 448
 - between paragraphs, 448
 - commands
 - `\bigskip`, 42, 84
 - `\enlargethispage`, 80, 527, 530
 - `\medskip`, 84
 - `\smallskip`, 42, 84
 - `\smash`, 96, 204, 205
 - to avoid using, 524–525
 - `\vfill` (fill com.), 86
 - `\vphantom`, 85, 204
 - `\vskip`, 42
 - `\vspace`, 42, 83
 - `\vspace*`, 85
 - in boxes, 96
 - in indexes, 274, 499
 - in math mode, 203–205
 - in text, 42, 83–85, 419, 520, 525, 529
 - preventing removal of, 85
 - stretchable, 450
 - using the `setspace` package, 78
 - `\vfill` (space fill com.), 86
 - viewers
 - PDF, *see under* PDF
 - viewing typeset files
 - PDF format, 6, 265
 - visible spaces (`\`), 9, 49
 - and `\verb*` command, 128
 - Visual
 - Guide, to multiline math formulas, 211
 - visual
 - design, 37, 251, 275–278, 353
 - guide, to multiline math formulas, 212
 - `Vmatrix` (subsid. math env.), 182, 240
 - `vmatrix` (subsid. math env.), 17, 18, 240
 - volume (BIBTeX database field), 468, 469
 - von, in bibliographic entries, 470
 - `\vphantom` (spacing com.), 85, 204
 - `\vskip` (spacing com.), 42
 - `\vspace` (spacing com.), 42, 83
 - `\vspace*` (spacing com.), 85
 - `\Vdash` (\Vdash binary rel.), 534
 - warning messages
 - about hyphenation, 76
 - Abstract should precede `\maketitle`
 - in AMS document classes, 255
 - Characters dropped after
 - `\end{verbatim}`, 127
 - `\end` occurred inside a group at level
 - x , 58
 - generated by overlapping braces, 58
 - Label(s) may have changed, 261
 - line numbers in, 12
 - lines are too wide, 11–76
 - Overfull `\hbox`, 12–76
 - suppressing, 76
 - recorded in log file, 36, 75
 - regarding font substitution, 139
 - Rerun to get cross-references right, 261
 - Warning--missing *field* in *label*, 488
 - when using `\NeedsTeXFormat`, 254, 427
 - with BIBTeX, 487–489
 - Warsaw (beamer theme), 29, 327, 328, 351
 - `\wedge` (\wedge math op.), 536

- weight, of fonts, *see under* font environments, font commands *and* font declarations
- white space
 - adjusting, [42](#), [525](#)
 - and binary operations and relations, [153](#)
 - between marginal notes, [448](#)
 - between paragraphs, [448](#)
 - commands, avoiding direct use of, [524–525](#)
 - determined by document classes, [81](#)
 - horizontal, [54](#), [81](#), [82](#), [85](#), [123](#), [152](#), [189–191](#), [228](#), [229](#), [408–409](#), [544](#), [550](#)
 - in math mode, [184](#), [238](#), [241](#)
 - in text mode, [9](#), [14](#), [50–53](#), [81–83](#), [85–86](#), [95](#), [106](#), [458](#)
 - in arguments of commands, [72](#)
 - in command definitions, [407](#)
 - in command names, [55](#)
 - in delimited commands, [420–421](#)
 - in math, [14](#), [39](#), [83–85](#), [152–153](#), [187–192](#), [203–205](#)
 - adjusting, [83](#)
 - in source files, [152](#)
 - in **tabular** environments, [115](#)
 - in text, [9](#), [81–86](#), [424](#), [471](#), [544](#), [550](#)
 - in **\verb** commands, [128](#)
 - in **verbatim** environments, [126](#)
 - suppressing, [424](#), [508](#)
 - terminating commands with, [55](#)
 - vertical, [81](#)
 - adding to table of contents, [520](#)
 - adjusting, [524–525](#)
 - in math mode, [83–85](#), [203–205](#)
 - in text mode, [42](#), [80](#), [83–86](#), [204](#), [419](#), [525](#), [529](#)
- \widehat** ($\hat{}$ math accent), [172](#), [173](#), [183](#), [543](#)
- \widetilde** ($\tilde{}$ math accent), [172](#), [173](#), [543](#)
- width
 - of columns in **tabular** environments, [116](#)
 - of fonts, *see* font width
 - of text blocks, [277](#), [443](#), [448](#)
 - of text boxes, [87–90](#), [449](#)
- \width** (length com.), [87](#), [88](#), [92](#)
- word processors, [9](#)
- words, are separated by spaces, [68](#)
- work** (folder), [5](#), [10](#), [12](#), [40](#), [176](#), [303](#), [316](#), [322](#), [327](#), [338](#), [339](#)
 - creating, [5](#)
- \wp** (\wp math sym.), [538](#)
- \wr** (\wr math op.), [536](#)
- wrapping, of lines by text editors, [70](#)
- Writing (GPT SP category), [554](#)
- xcb** (exercise env.), [522](#)
- xcolor** (pack.), [321](#), [344](#)
 - options, [344](#)
- \Xi** (Ξ Greek char.), [162](#), [532](#)
- \xi** (ξ Greek char.), [162](#), [532](#)
- \xleftarrow** (stretchable math arrow), [175](#)
- \xrightarrow** (stretchable math arrow), [175](#)
- xspace** (pack.), [408–409](#)
- xy-pic** (pack.), [247](#)
- \year** (time com.), [66](#)
- year** (BIB \TeX database field), [468](#), [469](#)
- zero, specifying in arguments, [95](#)
- \zeta** (ζ Greek char.), [162](#), [532](#)
- Zotero, [497](#)